

NPS53-78-002

NAVAL POSTGRADUATE SCHOOL

Monterey, California



SMOOTH SURFACE APPROXIMATION BY A LOCAL METHOD
OF INTERPOLATION AT SCATTERED POINTS

RICHARD FRANKE

Final Report for Period January - March 1978

Approved for Public Release; distributed unlimited

Prepared for: Chief of Naval Research
Arlington, VA 22217

FEDDOCS
D 208.14/2:NPS-53-78-002

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral Tyler F. Dedman
Superintendent

Jack Borsting
Provost

The work reported herein was supported by the Foundation Research Program of the Naval Postgraduate School with funds provided by the Chief of Naval Research.

Reproduction of all or part of this report is authorized.

This report was prepared by:

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NPS53-78-002	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Smooth Surface Approximation by a Local Method of Interpolation at Scattered Points		5. TYPE OF REPORT & PERIOD COVERED Final, Jan - Mar 1978
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Richard Franke		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Mathematics Naval Postgraduate School Monterey, CA 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61152N, RR 000-01-01, N0001478WR80023
11. CONTROLLING OFFICE NAME AND ADDRESS Chief of Naval Research Arlington, VA 22217		12. REPORT DATE April 1978
		13. NUMBER OF PAGES 55 54
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) scattered data, interpolation, surface approximation, optimal approximation, local methods		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report describes a computer program which constructs a surface passing through a set of data points $(x_k, y_k, f_k), k = 1, \dots, n$. It is based on previous work of the author (JIMA 19 (1977) 471-482), but uses a somewhat different approach which takes advantage of the nature of the approximations used and incorporates experience gained in the ensuing period. The surfaces are defined for all (x, y) points and have continuous second partial derivatives.		

1. The Interpolation Problem

Given the set of data points (x_k, y_k, f_k) , it is desired to construct a function $F(x, y)$, such that $F(x_k, y_k) = f_k, k = 1, \dots, n$. For large sets of data it is desirable for the method to be local, that is, the value of $F(x, y)$ depends only on the value of f_k at nearby points (x_k, y_k) .

This problem is receiving a great deal of attention and discussions of it and proposed methods can be found in [1], [2], [4], [7], and [8].

2.0. The Interpolation Scheme

This is a local method, the general idea having been discussed in [4]. The basic idea is to construct local interpolants, F_ℓ , which are then weighted by functions W_ℓ having limited support to obtain the function

$$(1) \quad F(x, y) = \sum_{\ell} W_{\ell}(x, y) F_{\ell}(x, y) / \sum_{\ell} W_{\ell}(x, y) .$$

The details are fully discussed in the reference, but the important fact is that $F(x, y)$ will take on the value f_k at (x_k, y_k) if for each ℓ where $W_{\ell}(x_k, y_k) \neq 0$, $F_{\ell}(x_k, y_k) = f_k$. In the referenced paper, the weight functions W_{ℓ} were taken to be of the form

$$W_{\ell}(x, y) = \begin{cases} 1 - 3(d_{\ell}/r_{\ell})^2 + 2(d_{\ell}/r_{\ell})^3 & , \quad d_{\ell} \leq r_{\ell} \\ 0 & , \quad d_{\ell} > r_{\ell} \end{cases}$$

where r_{ℓ} is the radius of the smallest circle centered at (x_{ℓ}, y_{ℓ}) which contains a given fixed number of data points, and d_{ℓ} is the distance from (x, y) to (x_{ℓ}, y_{ℓ}) .

This scheme, used with local interpolants, F_{ℓ} which were taken to be certain optimal approximations, yielded reasonably good results. However, the computational burden was rather high. This was due in part to a great deal of overlap in the regions where the W_{ℓ} are nonzero. In addition, the approxi-

mation is defined only on the union of the circles around each of the data points, which could cause problems.

One of the advantages of using optimal approximations is that the basis functions are generated by the (x_k, y_k) and the system is automatically nonsingular. Within some limitations, the resulting equations can easily be solved by Cholesky decomposition [5]. This opens the way for the present approach, which is to choose rectangular regions on which weight functions are non-zero, thus being able to carefully govern the amount of overlap with a resulting decrease in the necessary computations, as well as simplification of the weight functions.

2.1. Weight Functions

With these ideas in mind, we are now ready to describe the present selection of regions over which the weight functions are non-zero. These regions will be rectangles defined by the following parameters. Let n_x and n_y be given positive integers and let finite values of $\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_{n_x}$ and $\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_{n_y}$ be given. For convenience in this section we let $\tilde{x}_0 = \tilde{y}_0 = -\infty$ and $\tilde{x}_{n_x+1} = \tilde{y}_{n_y+1} = \infty$. For each $i = 1, \dots, n_x$ and $j = 1, \dots, n_y$, let R_{ij} denote the rectangle $(\tilde{x}_{i-1}, \tilde{x}_{i+1}) \times (\tilde{y}_{j-1}, \tilde{y}_{j+1})$.

Let $H_5(s) = 1 - s^3(6s^2 - 15s + 10)$, the Hermite quintic satisfying $H_5(0) = 1, H_5'(0) = H_5''(0) = H_5(1) = H_5'(1) = H_5''(1) = 0$. We then define functions which are piecewise quintics with continuous second derivatives which have the property that they are non-zero on two intervals and satisfy

$$V_i(x_j) = \delta_{ij}, \quad i = 1, \dots, n_x, \quad j = 0, 1, \dots, n_x + 1$$

$$U_j(y_i) = \delta_{ji}, \quad j = 1, \dots, n_y, \quad i = 0, 1, \dots, n_y + 1.$$

In particular,

$$V_1(x) = \begin{cases} 1 & , x < \tilde{x}_1 \\ H_5\left(\frac{x - \tilde{x}_1}{\tilde{x}_2 - \tilde{x}_1}\right) & , \tilde{x}_1 \leq x < \tilde{x}_2 \\ 0 & , x \geq \tilde{x}_2 \end{cases}$$

$$V_i(x) = \begin{cases} 0 & , x < \tilde{x}_{i-1} \\ 1 - V_{i-1}(x) & , \tilde{x}_{i-1} \leq x < \tilde{x}_i \\ H_5\left(\frac{x - \tilde{x}_i}{\tilde{x}_{i+1} - \tilde{x}_i}\right) & , \tilde{x}_i \leq x < \tilde{x}_{i+1} \\ 0 & , x \geq \tilde{x}_{i+1} \end{cases}$$

for $i = 2, \dots, n_x - 1$, and

$$V_{n_x}(x) = \begin{cases} 0 & , x < \tilde{x}_{n_x-1} \\ 1 - V_{n_x-1}(x) & , \tilde{x}_{n_x-1} \leq x < \tilde{x}_{n_x} \\ 1 & , x \geq \tilde{x}_{n_x} \end{cases}$$

The $U_j(Y)$ are dual. Then, if we define

$$W_{ij}(x,y) = V_i(x)U_j(Y), \quad i = 1, \dots, n_x, \quad j = 1, \dots, n_y,$$

it is easily observed that the function $W_{ij}(x,y)$ has support $Cl(R_{ij})$ and that the functions form a partition of unity for the plane, i.e.,

$$\sum_{i,j} W_{ij}(x,y) \equiv 1 \quad \text{for all } (x,y).$$

These properties allow the construction of the interpolation function (1) to proceed easily since any point (x,y) is in at most four R_{ij} , and the denominator of (1) is always $\equiv 1$, allowing us to write

$$F(x,y) = \sum_{i,j} W_{ij}(x,y)Q_{ij}(x,y).$$

We again emphasize that at most four terms in the sum are non-zero.

The appropriate choice of x_i and y_j as well as n_x and n_y depend on the data as well as the choice of local interpolating functions $Q_{ij}(x,y)$. For this reason we defer discussion of the selection of these grid lines until after we discuss the choice of $Q_{ij}(x,y)$.

2.2. Local Interpolation Functions

The only restriction on the local interpolation functions $Q_{ij}(x,y)$ are that they interpolate all data points in R_{ij} and that they are defined for all (x,y) in R_{ij} . Polynomials sometimes fail to satisfy these conditions. The use of optimal approximations in Sard corner spaces has been investigated [5], and for small numbers of data points, the approximations can be computed in straightforward fashion. One possible defect in such approximations is their lack of polynomial precision: even constants are not approximated exactly. With only a slight complication this can be overcome, since by a theorem of Barnhill and Gregory [3], the boolean sum operator $B \oplus L$ has the interpolation properties of B and the function precision of L . Here we are thinking of Bf as the optimal approximation in $B_{[2,2]}$ while Lf is the least squares fit by a linear function.

The implemented version of the program embodies three options: (1) Use optimal approximations in $B_{[2,2]}$ as the local interpolation functions; (2) Use the least squares linear approximation instead of an interpolation function and (3) Use the optimal approximation in $B_{[2,2]}$ boolean sum the least squares linear approximation. The second option yields a surface which in general does not interpolate the given data. The third option is achieved computationally as $(B \oplus L)f = (B + L - BL)f = Lf + B(f - Lf)$.

The use of the boolean sum has a desirable effect in that it removes much of the effect of linear transformations of the data on the overall approximation.

However, for complete consistency with respect to translation and change of the measure of distance, each rectangle

$$[\tilde{x}_{i-1}, \tilde{x}_{i+1}] \times [\tilde{y}_{j-1}, \tilde{y}_{j+1}]$$

is transformed to the unit square for the optimal approximation. For these purposes, we take $\tilde{x}_0 = \min_k x_k$ and $\tilde{x}_{n+1} = \max_k x_k$, and the dual in y . The base point (a,b) is taken to be $(0,0)$ for all $i,j \geq 2$, while it is taken to be $(1,1)$ for $i = j = 1$, $(0,1)$ for $i \geq 2, j = 1$, and $(1,0)$ for $i = 1, j \geq 2$. This yields lines of discontinuity in the second derivatives which are nowhere interior to the support regions for the weight functions W_{ij} , thus assuring continuous second derivatives in the overall approximation.

The overall approximation is invariant with respect to linear transformations which leave the directions of the axes unchanged. Since lines of discontinuities in the third derivatives occur along horizontal and vertical lines the approximation is not invariant with respect to rotations.

The points associated with R_{ij} include all the points in the closure of R_{ij} . Because approximation by a linear function requires at least three points, a parameter MINPTS, is used to assure that at least MINPTS points are selected for each R_{ij} . If extra points are required, they are taken as the closest points in the sup norm, distance being measured after

$$[x_{i-1}, x_{i+1}] \times [y_{j-1}, y_{j+1}]$$

has been transformed onto $[0,1]^2$. Presently MINPTS is set to three and this has been satisfactory. It is easily changed, if desired or necessary. For example, if some R_{ij} has only three colinear points associated with it, the scheme will fail under options (2) or (3). Then one must either increase the value of MINPTS or use option (1).

2.3. Selection of Grid Lines

It is desirable to have automatic selection of grid lines, that is, values of \tilde{x}_i and \tilde{y}_j . This should be accomplished in some manner which results in rectangles R_{ij} which contain approximately equal numbers of points. For data which is poorly distributed this may not be possible. However, for somewhat uniformly distributed points the process we describe here works quite well.

The selection of the grid lines is determined by one parameter, called NPPR, for "number of points per rectangle." The grid lines are then chosen so that there will be approximately NPPR points in each rectangle, R_{ij} . If there are additional points added to certain rectangles to make up MINPTS points the average may be higher. The average is, of course, dependent on the data set.

Equal numbers of grid lines are chosen in each direction, that is $n_x = n_y$. Because we want NPPR points per rectangle, each subrectangle

$$(\tilde{x}_i, \tilde{x}_{i+1}) \times (\tilde{y}_j, \tilde{y}_{j+1})$$

should have $\frac{1}{4}$ NPPR points. Thus we want to choose $n_x = n_y$ so that $(n_x + 1)^2 \cdot \frac{1}{4} \text{NPPR} = n$, the total number of data points. Thus, we take n_x to be the nearest integer to $(4n/\text{NPPR})^{1/2} - 1$.

Grid lines, that is \tilde{x}_i and \tilde{y}_j values, are now determined by choosing these values so that approximately $n/(n_x + 1)$ points occur in each $(\tilde{x}_i, \tilde{x}_{i+1}]$ and each $(\tilde{y}_j, \tilde{y}_{j+1}]$. Specifically, let \hat{x}_k denote the values of x_k given in nondecreasing order, then $x_i = \hat{x}_k$, where k is the integer nearest $in/(n_x + 1)$ for $i = 1, 2, \dots, n_x$. The selection in y is dual.

3.0. Implementation

The scheme is implemented in a set of subprograms, only one of which is normally referenced by the user. The hierarchy of subprograms is given in figure 1. A brief description of them, according to level, follows.

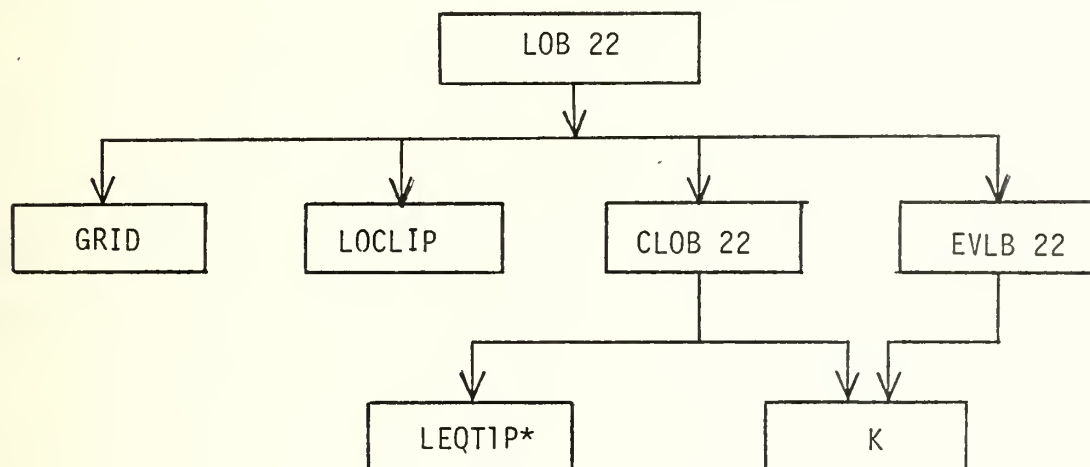


Figure 1

3.1.1. User's Program

The user's program must provide the data points, (x_k, y_k, f_k) , $k = 1, \dots, n$, as well as the points $x0_i$ and $y0_j$ for the grid of points at which the interpolation function is to be evaluated. In addition the user's program must provide workspace arrays, IWK and WK, and an array F0 for the returned function values. The amount of storage required in the arrays IWK and WK are not known a priori, but are estimated as follows.

For IWK, approximately $4n(1 + 1/NPPR)$ locations are required. This has generally proven to be an overestimate. For WK, the storage required depends on the type of local approximation used and is approximately:

*This is a Cholesky decomposition equation solver in the IMSL Library, which itself references several other subroutines from the IMSL Library.

for MODE = 1, $4(n + \sqrt{n/NPPR})$;
for MODE = 2, $4(3n/NPPR + \sqrt{n/NPPR})$; and
for MODE = 3, $4(n + 3n/NPPR + \sqrt{n/NPPR})$.

As with the estimates for IWK, these estimates have proven to generally be overestimates. The precise number of storage locations required in each array are returned to the user's program by the principal subroutine.

Under the usual option, the user specifies $NPPR > 0$. If the user wishes to specify the grid lines, NPPR is set to zero, and additional input in the arrays IWK and WK is necessary. In the array IWK, the user specifies n_x in IWK(1) and n_y in IWK(2). The grid lines are specified in the array WK, according to the following.

WK(1) is $\min_k x_k$,
WK(2),...,WK($n_x + 1$) are the vertical grid values $\tilde{x}_1, \dots, \tilde{x}_{n_x}$, in increasing order,
WK($n_x + 2$) is $\max_k x_k$,
WK($n_x + 3$) is $\min_k y_k$,
WK($n_x + 4$),...,WK($n_x + n_y + 3$) are the horizontal grid values $\tilde{y}_1, \dots, \tilde{y}_{n_y}$ in increasing order, and
WK($n_x + n_y + 4$) is $\max_k y_k$.

3.2.1. Subroutine LOB 22

This subroutine provides the interface between the user and the set of subroutines which implement the method. Generally LOB 22 sets up storage locations in the arrays IWK and WK, determines parameters required by other subroutines, and calls subroutines to (1) generate the grid (if necessary), (2) determine the interpolation points for each rectangle, (3) compute coefficients for the local interpolating functions, and finally, (4) to evaluate the interpolation function on the desired grid of points.

3.3.1. Subroutine GRID

This subroutine selects the values of \tilde{x}_i and \tilde{y}_j in accordance with the discussion in Section 2.3.

3.3.2. Subroutine LOCLIP

This subroutine determines the local interpolation points for each R_{ij} in accordance with the last paragraph of Section 2.2.

3.3.3. Subroutine CLOB 22

This subroutine computes the coefficients in the least squares plane (MODE = 2 or 3) and the coefficients in the optimal approximation (MODE = 1 or 3) or each of the rectangles R_{ij} . In the present implementation the IMSL Cholesky decomposition equation solver LEQT1P is used. This could be replaced at a facility where IMSL is not available, although according to IMSL policy, LEPT1P (and associated subroutines) can be used as part of this package at any facility. Because of the short single precision word length on the IBM 360/370 series computers, on which this program was implemented, the coefficients for the system of equations for the optimal approximation are generated in double precision. On computers with a longer word length, the double precision variables in this routine can be safely removed. Other double precision statements occur in EVLB 22 and function K, which must also be removed.

3.3.4. Subroutine EVLB 22

This subroutine evaluates the approximation (2) on the set of points $(x0_i, y0_j)$ as specified by the user, and returns the values in F0. As noted above, the double precision variables in this subroutine should be removed on computers with longer word length than the IBM 360/370 series.

3.4.1. Function K

This function evaluates the representers for point evaluation functionals in $B_{[2,2]}$. For evaluation at (u,v) , base point at (a,b) , the representer, as a function of (s,t) [6] is

$$K(a,b;u,v,s,t) = g_2(a;u,s)g_2(b;v,t), \text{ where}$$

$$g_2(a;u,s) = G_2(a;u,s) = (s-u)_+^{(3)} + 1 + (u-a)(s-a) + (u-a)(s-a)_+^{(2)} + (s-a)_+^{(3)}$$

for $a \leq u$, and

$$g_2(a;u,s) = G_2(-a;-u,-s) \text{ for } u < a.$$

The arguments of this function are all single precision, but because of the short word length of the IBM 360/370 computers, all calculations are performed in double precision, and the returned value is double precision. On computers with longer word lengths these calculations can be done in single precision.

4.0. Examples

The method has been applied to a number of sets of data with good results. Figures 3 - 5 show test surfaces and results of applying the method for each of the options for local approximations. The three surfaces are described by

$$(C) \quad F(x,y) = \tanh(y - x) + 1,$$

$$(S) \quad F(x,y) = 3/2[\cos(3/5(y - 1)) + 5/4]/[1 + (\frac{x-4}{3})^2], \text{ and}$$

$$(E) \quad F(x,y) = 9\{3/4 \exp(\frac{-(x-3)^2 - (y-3)^2}{4}) + \exp(-(\frac{x}{7})^2 - (\frac{y}{10}))$$

$$- \frac{1}{5} \exp(-(x - 5)^2 - (y - 8)^2)$$

$$+ \frac{1}{2} \exp(\frac{-(x-8)^2 - (y-4)^2}{4})\},$$

respectively. The 100 interpolation points were chosen at random within a unit square centered at (i,j) for $i,j = 1,2,\dots,10$. The points are shown in Figure 2 as +',s, with the convex hull shown by dashed lines, while the square

$(1,10)^2$, on which the resulting interpolation functions were evaluated, is given by the solid lines. The diagonal line shows the direction toward the viewing point.

There does not appear to be a great deal of difference between the optimal approximation and the optimal approximation boolean sum least squares plane. Generally the latter option has slightly smaller errors and slightly less noticeable defects. Gross defects in the approximations can generally be traced to a lack of data in that particular part of the region.

The effect of varying the parameter NPPR is shown in Figures 6 - 14. Some general observations are possible from this set of views. Most apparent is the fact that option (2), the least squares plane fit as the local approximation does not appear to lead to very good results. In general, however, the smaller value of NPPR gives better results, visually, and usually better accuracy, too.

The choice of NPPR = 6 for options (1) and (3) appears to be a reasonable one. For surfaces with sharp gradients, as in Figure 6, it appears that localizing the behavior as much as possible with a smaller value of NPPR is the best strategy. For smooth surfaces, such as in Figure 9 and 12 it appears the opposite is true, where NPPR = 8 seems to lead to the best results.

The storage and timing results are given in Table 1. The storage refers to requirements of the two workspace arrays provided by the user. The timing is for calculation of the 1089 points generated for the plots. The program was run under the Fortran H compiler on the IBM 360 model 67 at the Naval Postgraduate School. Computation times are dependent on external factors and may vary from run to run.

5.0. Acknowledgements

During the first half of 1977 the author was a Visiting Associate Professor at the University of Utah. Interactions with Professor R. E. Barnhill and his students on the subject of surface approximation proved to be fruitful. The kernel of a number of ideas in the present scheme germinated during that time. Thanks also go to Rosemary E. Chang of Sandia Laboratories (Livermore) who first undertook to run the program on a CDC computer. Improvements in the program description and the test program were a result of those efforts.

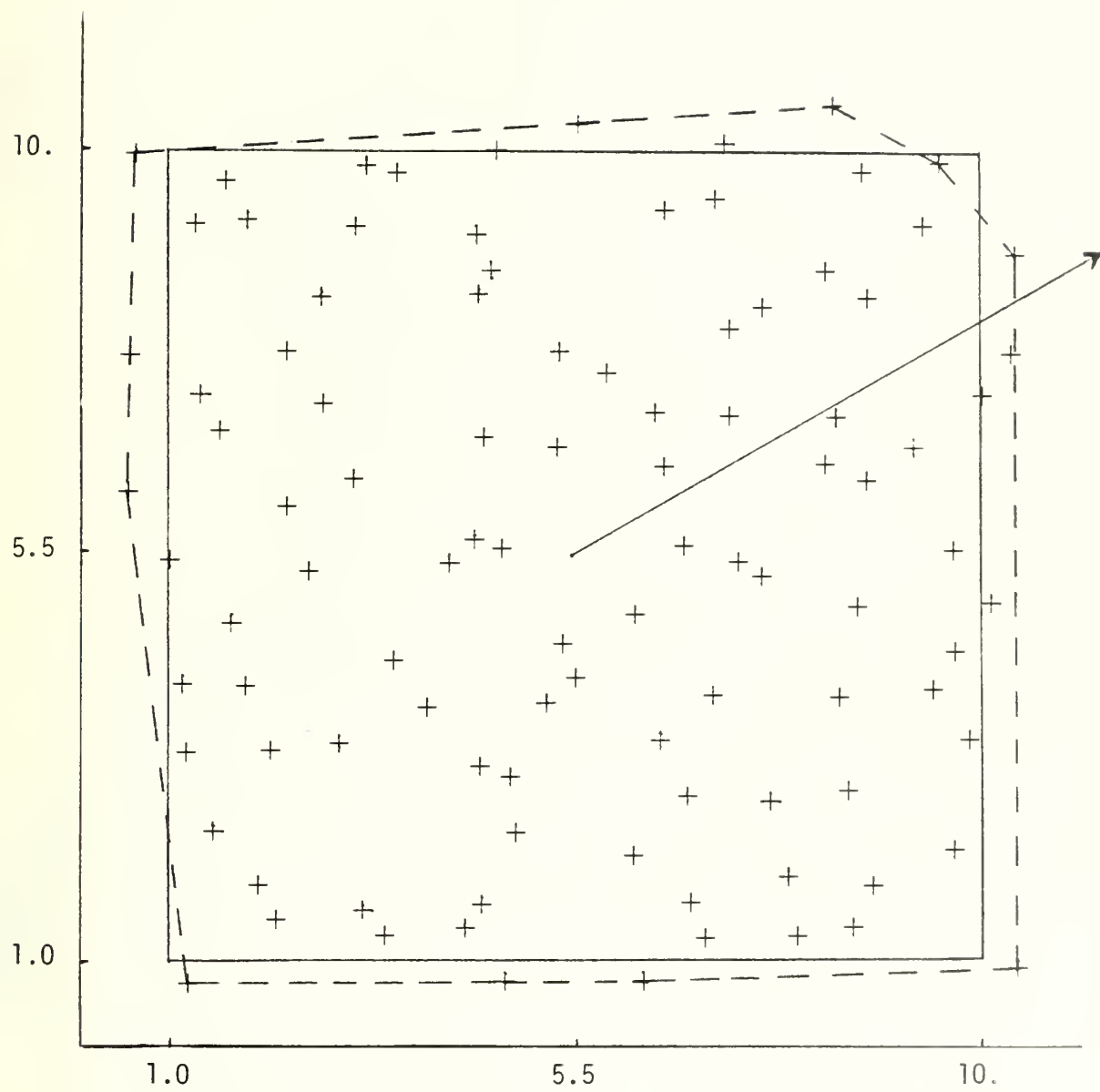
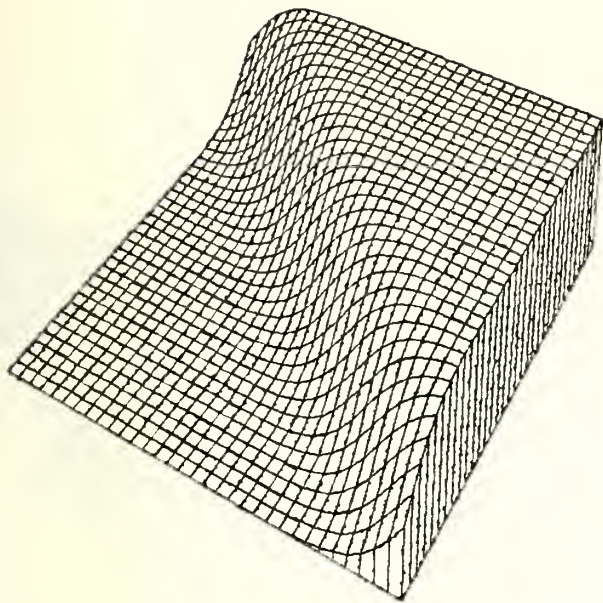


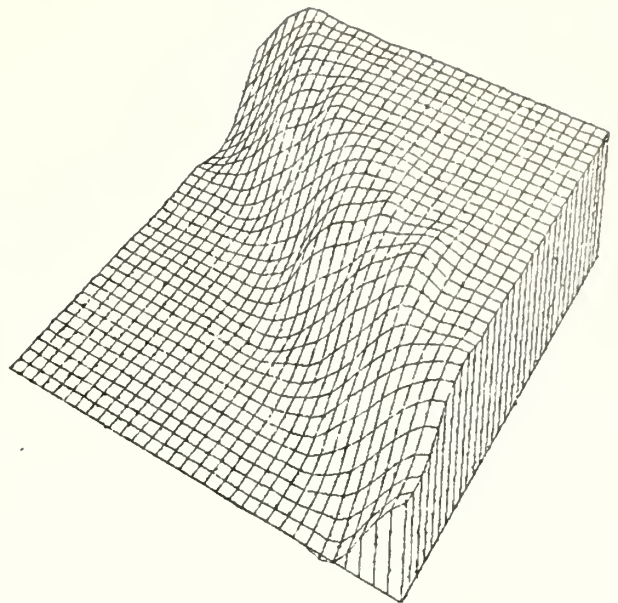
Figure 2

NPPR	MODE	NIWK	NWK	Approx. Time (sec.)
4	1	437	375	7.9
4	2	437	265	1.3
4	3	437	618	8.2
6	1	380	346	10.9
6	2	380	165	1.1
6	3	380	493	11.1
8	1	351	328	12.9
8	2	351	124	1.0
8	3	351	436	13.1

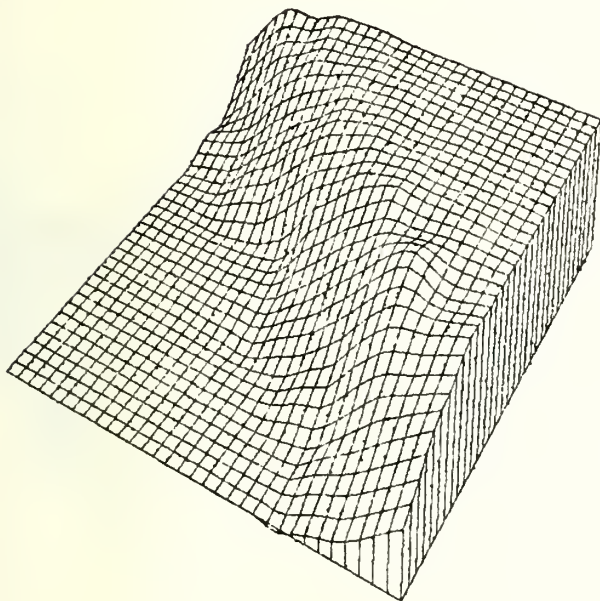
Table 1



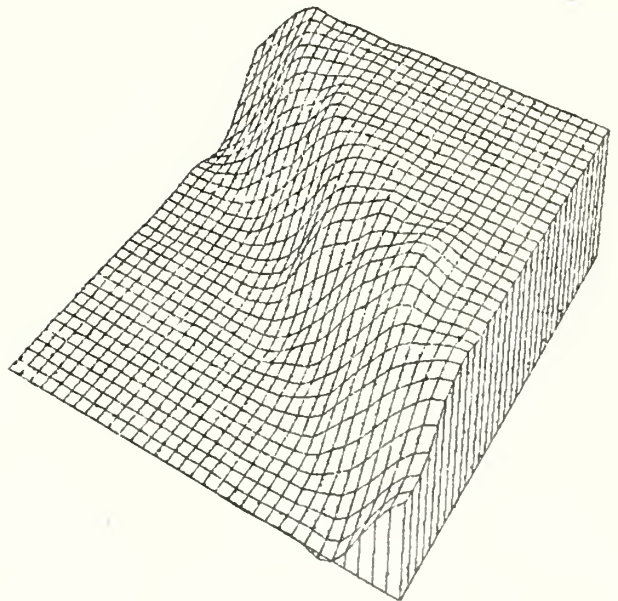
Test Surface
Cliff Function



Mode = 1 , $E_{\max} = .468$
 $E_{\text{rms}} = .0263$
 $E_{\text{mean}} = .0526$

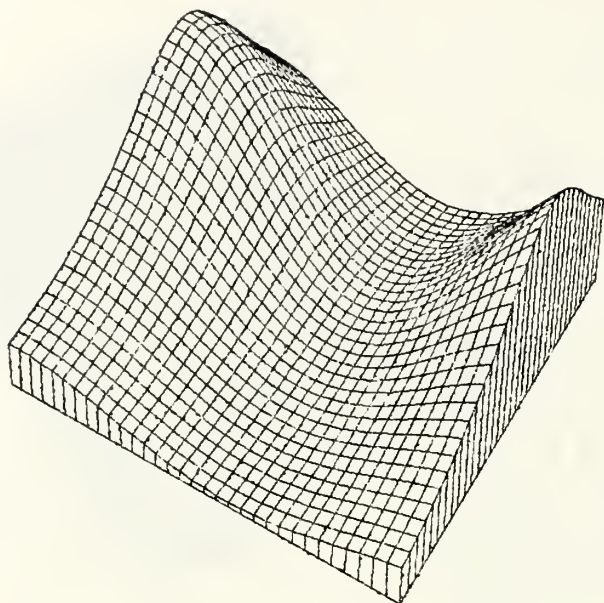


Mode = 2 , $E_{\max} = .283$
 $E_{\text{rms}} = .0523$
 $E_{\text{mean}} = .0864$

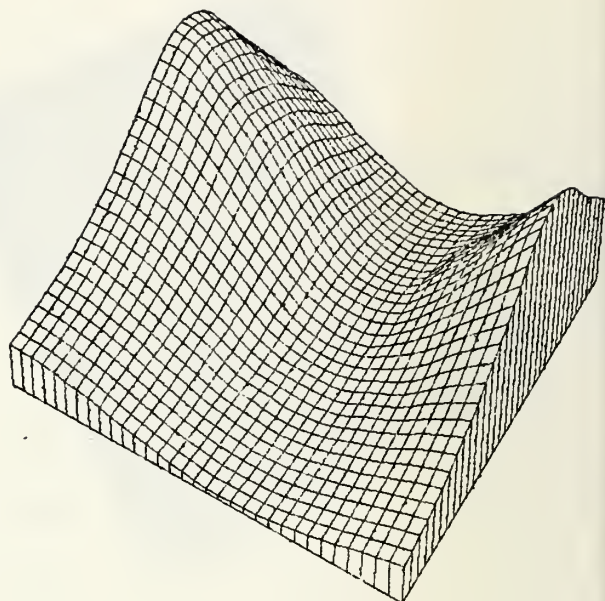


Mode = 3 , $E_{\max} = .466$
 $E_{\text{rms}} = .0257$
 $E_{\text{mean}} = .0527$

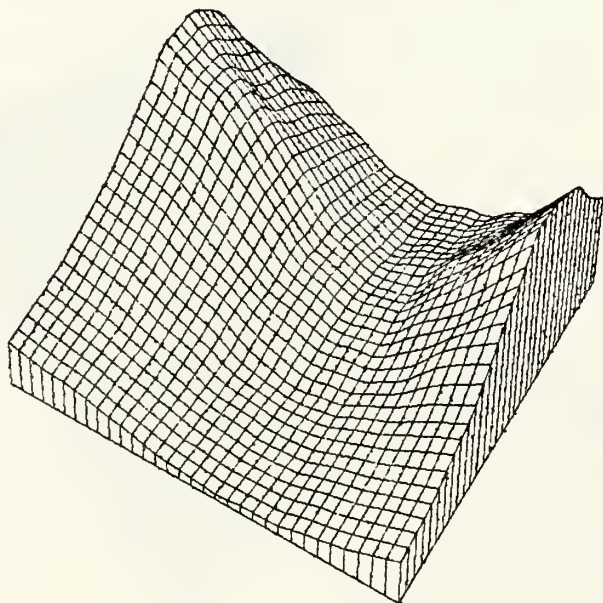
Figure 3 (NPPR = 6)



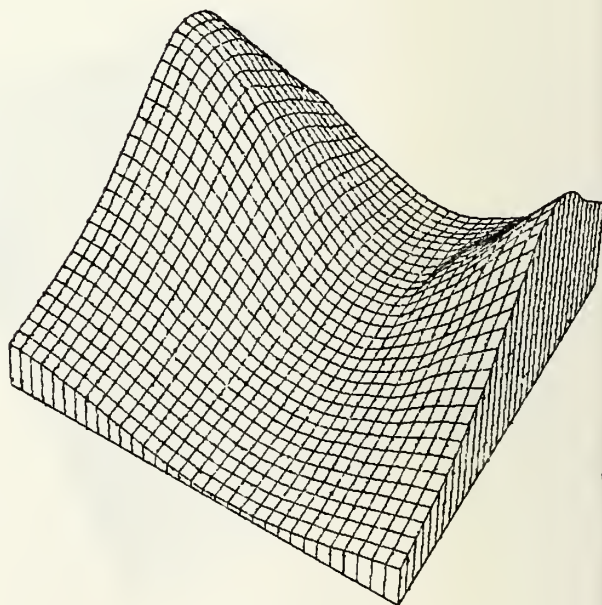
Test Surface
Saddle Function



Mode = 1 , $E_{\max} = .187$
 $E_{\text{rms}} = .0156$
 $E_{\text{mean}} = .0273$

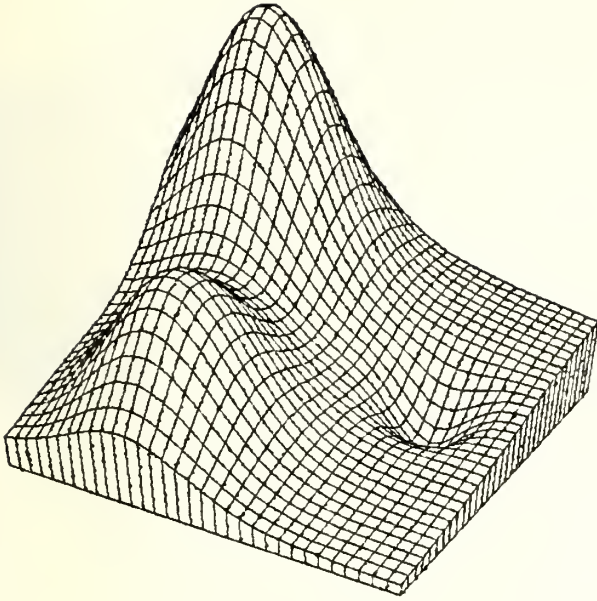


Mode = 2 , $E_{\max} = .389$
 $E_{\text{rms}} = .0495$
 $E_{\text{mean}} = .0739$

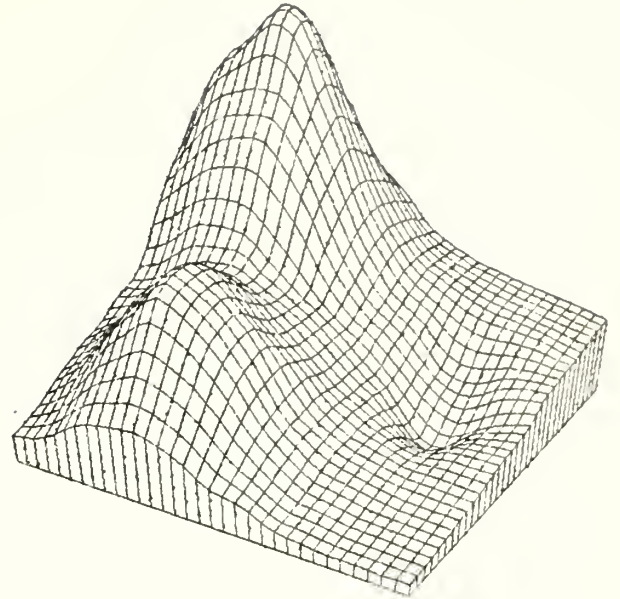


Mode = 3 , $E_{\max} = .178$
 $E_{\text{rms}} = .0148$
 $E_{\text{mean}} = .0265$

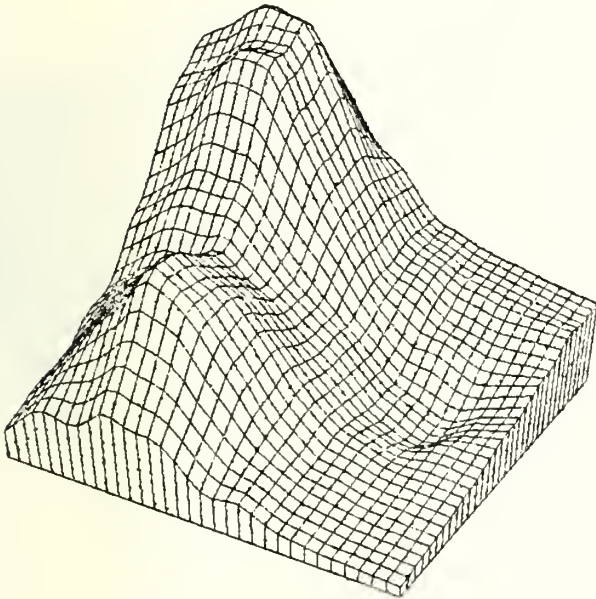
Figure 4 (NPPR = 6)



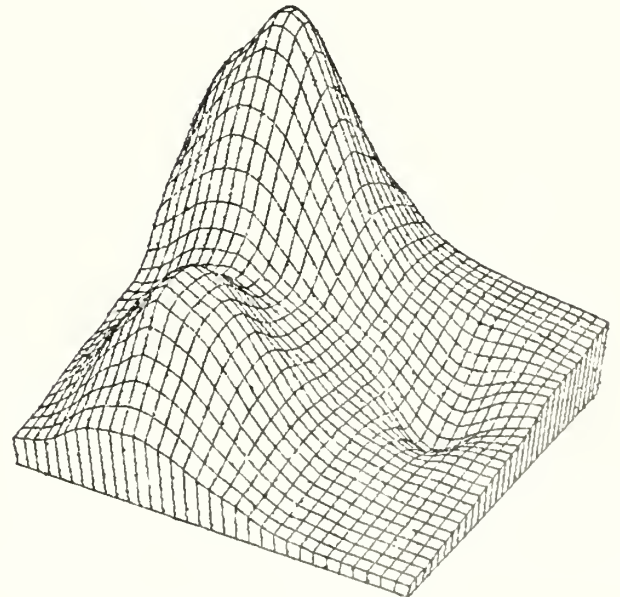
Test Surface
Exponentials



Mode = 1 , $E_{\max} = .974$
 $E_{\text{rms}} = .0929$
 $E_{\text{mean}} = .169$

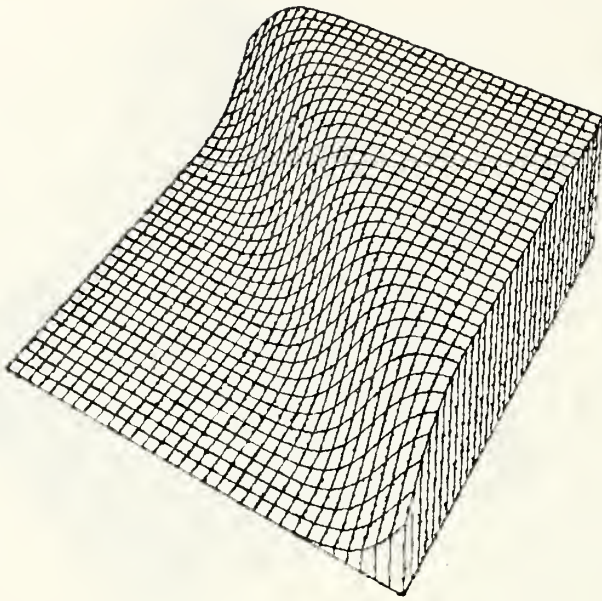


Mode = 2 , $E_{\max} = .216$
 $E_{\text{rms}} = .209$
 $E_{\text{mean}} = .366$

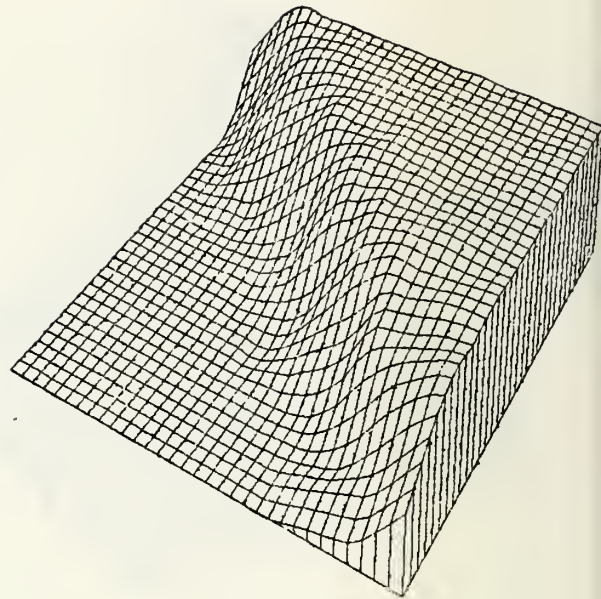


Mode = 3 , $E_{\max} = .827$
 $E_{\text{rms}} = .0757$
 $E_{\text{mean}} = .133$

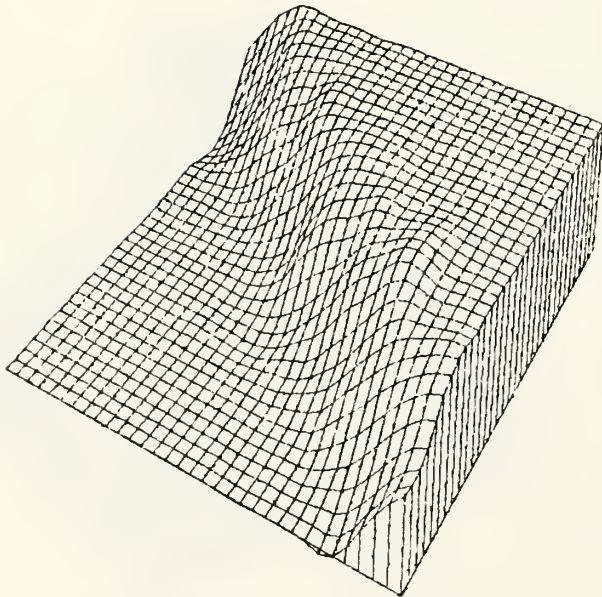
Figure 5 (NPPR = 6)



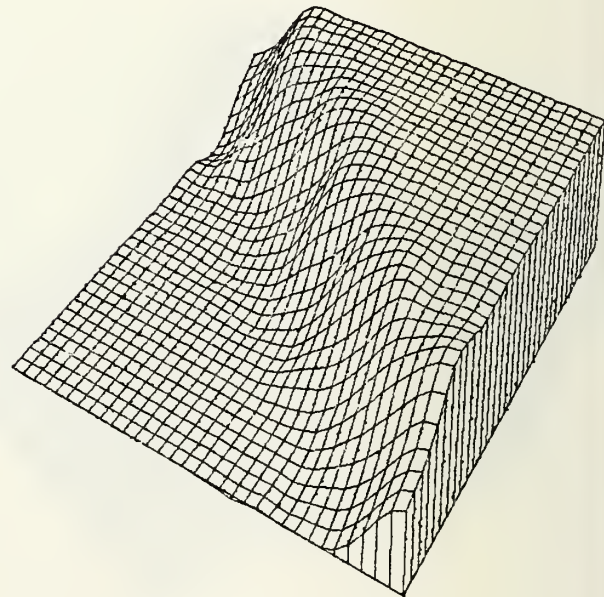
Test Surface
Cliff Function



NPPR = 4 , $E_{\max} = .265$
 $E_{\text{rms}} = .0271$
 $E_{\text{mean}} = .0513$

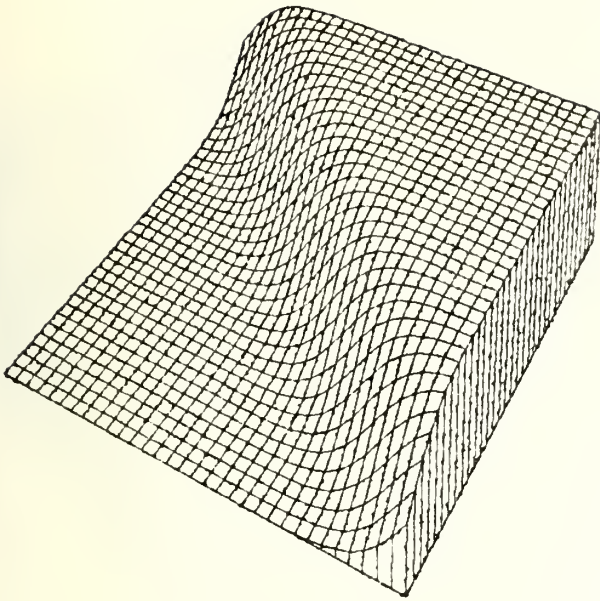


NPPR = 6 , $E_{\max} = .466$
 $E_{\text{rms}} = .0257$
 $E_{\text{mean}} = .0527$

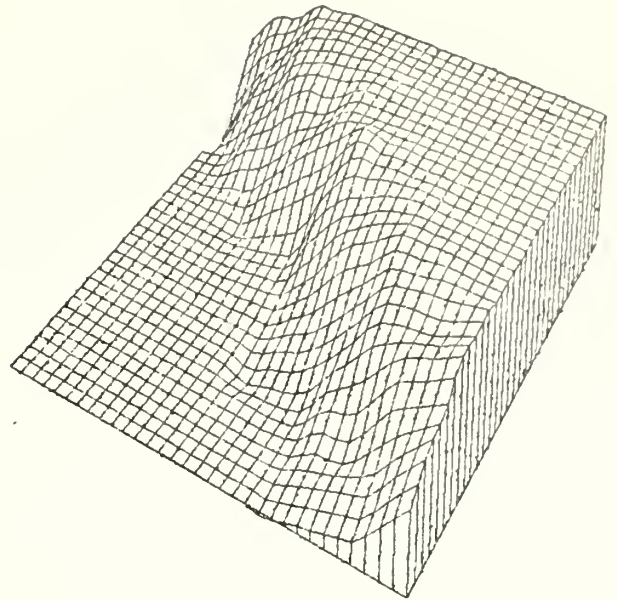


NPPR = 8 , $E_{\max} = .467$
 $E_{\text{rms}} = .0308$
 $E_{\text{mean}} = .0633$

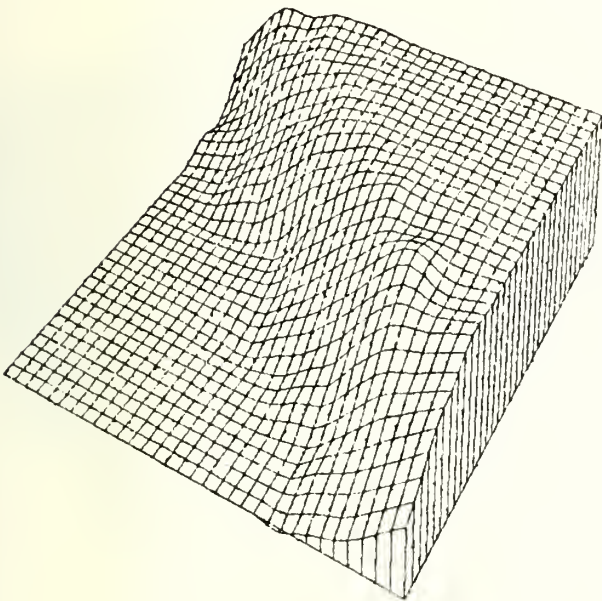
Figure 6 (Mode = 1)



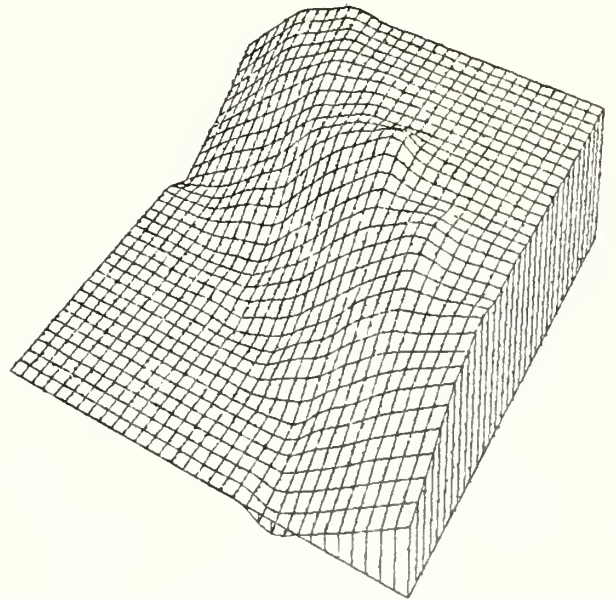
Test Surface
Cliff Function



NPPR = 4 , E_{\max} = .336
 E_{rms} = .0435
 E_{mean} = .0797

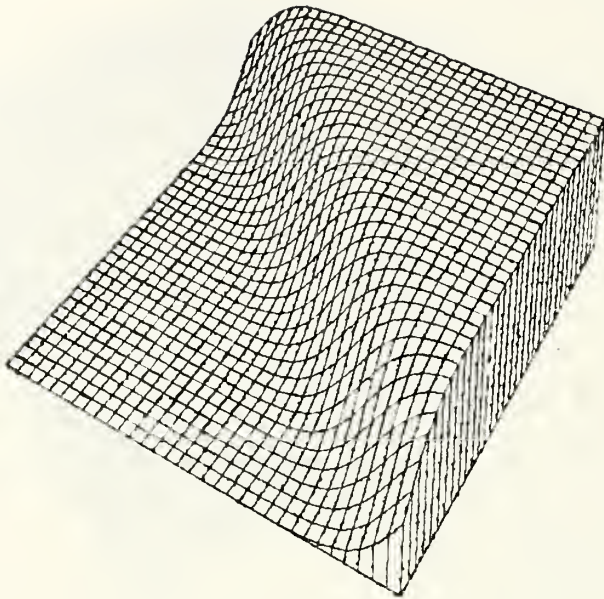


NPPR = 6 , E_{\max} = .283
 E_{rms} = .0523
 E_{mean} = .0864

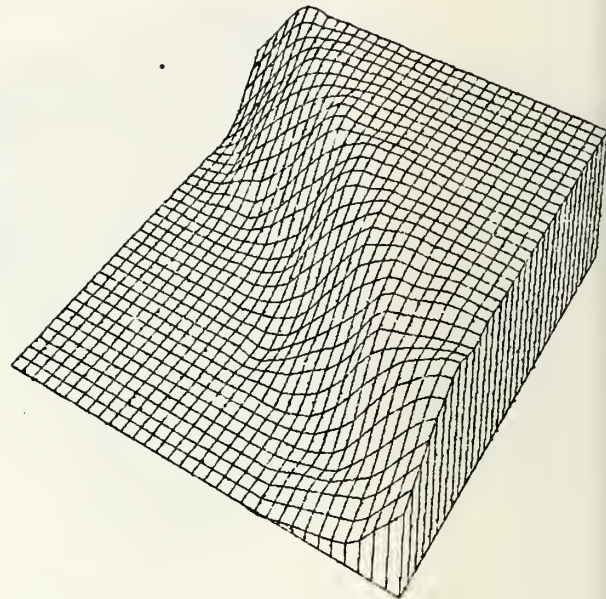


NPPR = 8 , E_{\max} = .375
 E_{rms} = .0692
 E_{mean} = .113

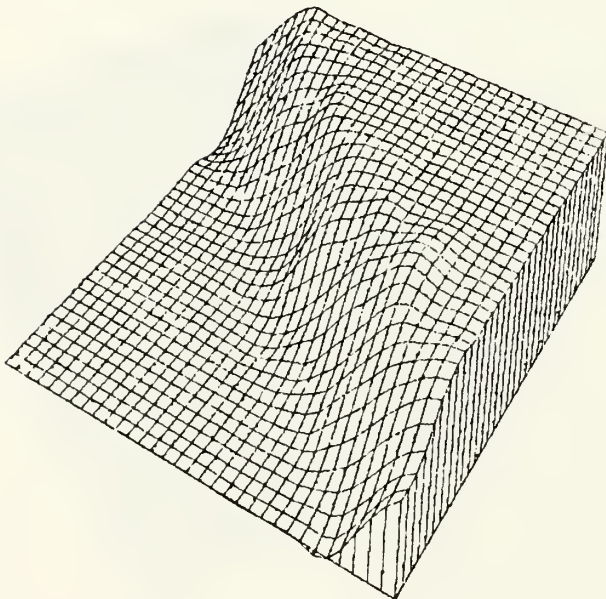
Figure 7 (Mode = 2)



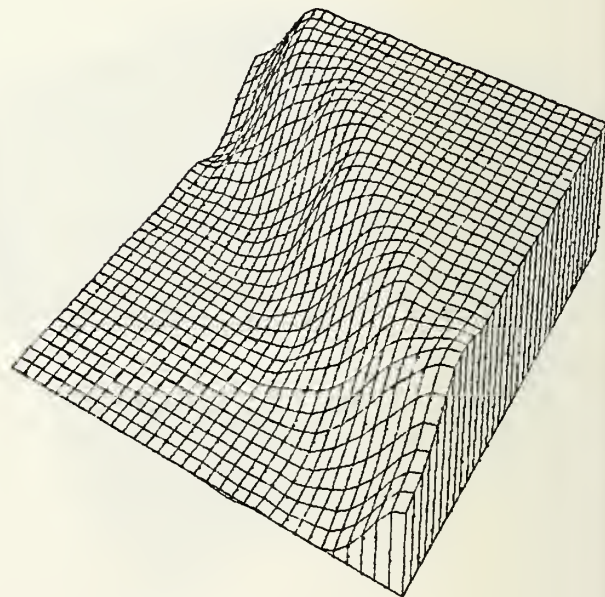
Test Surface
Cliff Function



NPPR = 4 , $E_{\max} = .261$
 $E_{\text{rms}} = .0246$
 $E_{\text{mean}} = .0500$

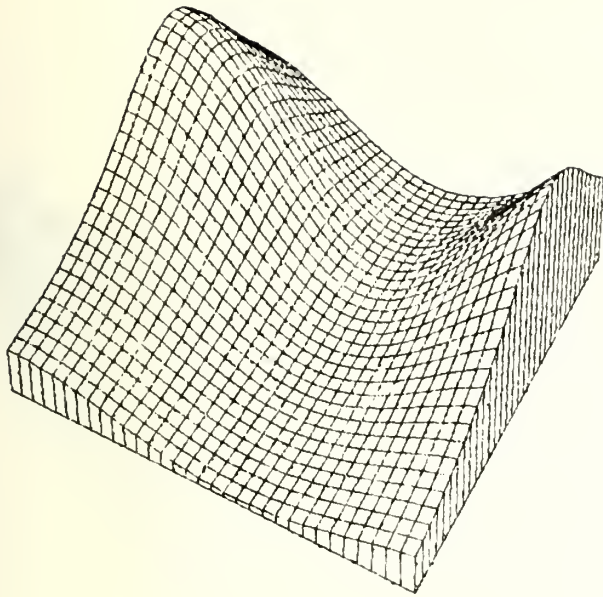


NPPR = 6, $E_{\max} = .468$
 $E_{\text{rms}} = .0263$
 $E_{\text{mean}} = .0526$

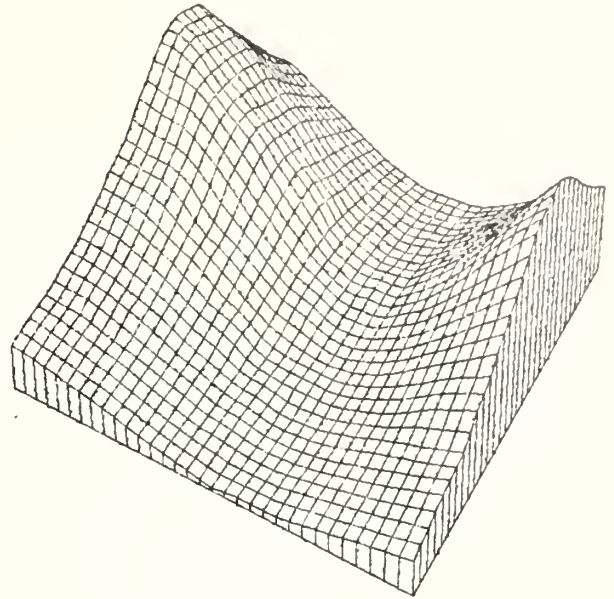


NPPR = 8 , $E_{\max} = .462$
 $E_{\text{rms}} = .0304$
 $E_{\text{mean}} = .0622$

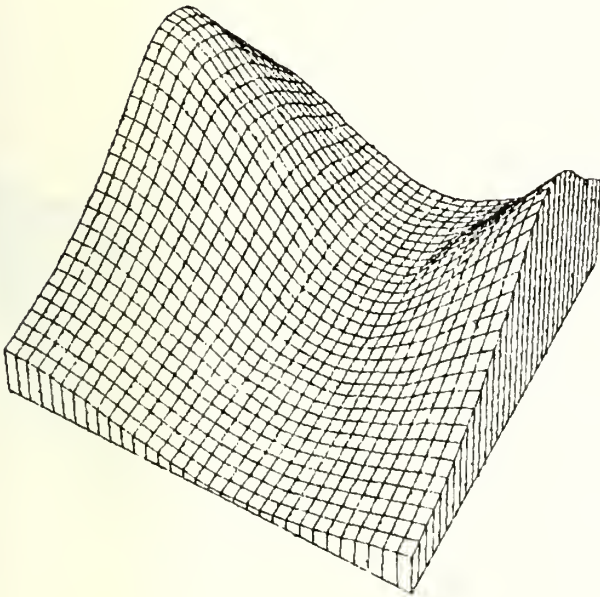
Figure 8 (Mode = 3)



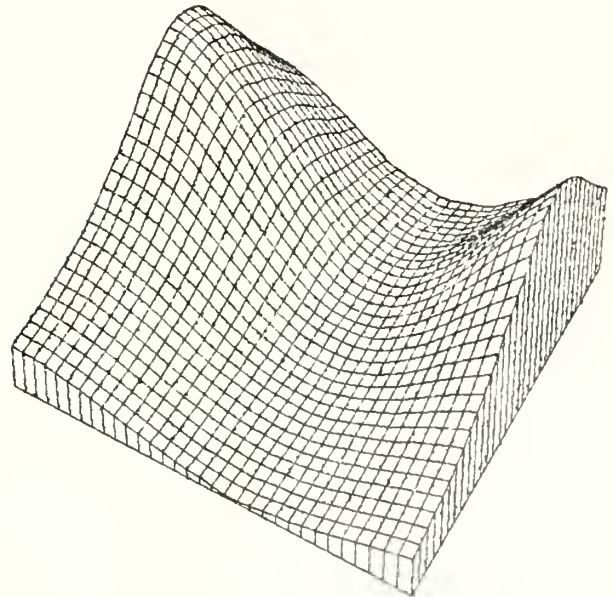
Test Surface
Saddle Function



NPPR = 4 , $E_{\max} = .208$
 $E_{\text{rms}} = .0249$
 $E_{\text{mean}} = .0398$

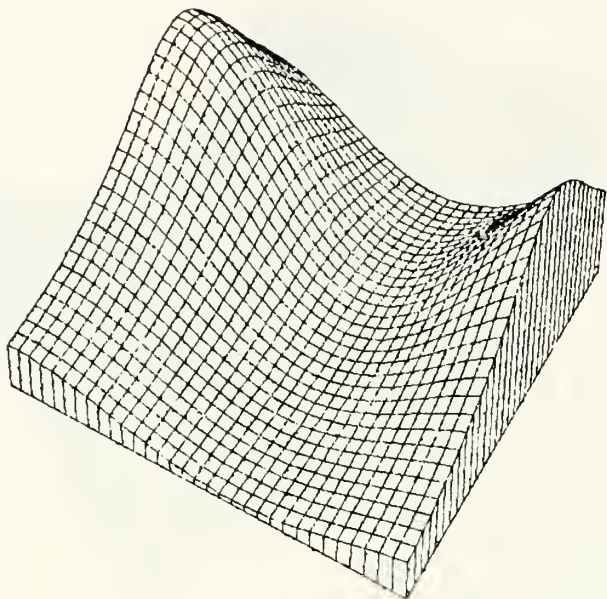


NPPR = 6 , $E_{\max} = .187$
 $E_{\text{rms}} = .0156$
 $E_{\text{mean}} = .0273$

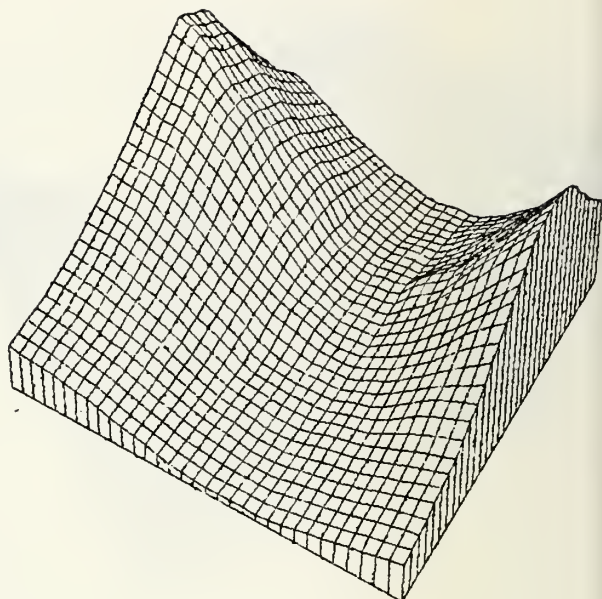


NPPR = 8 , $E_{\max} = .154$
 $E_{\text{rms}} = .0118$
 $E_{\text{mean}} = .0211$

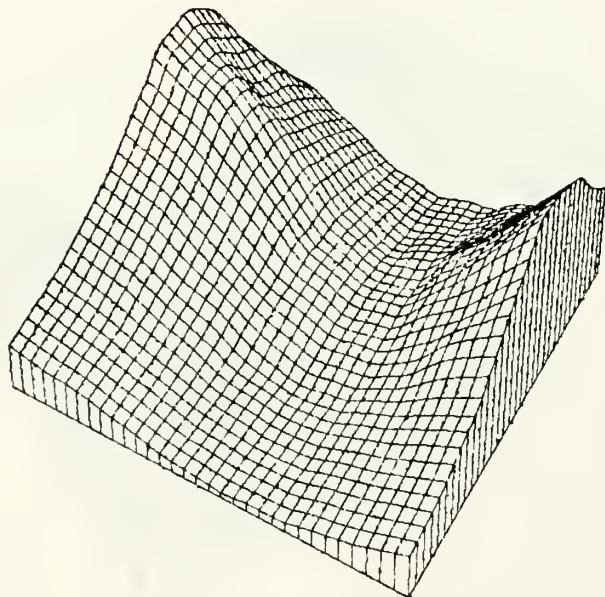
Figure 9 (Mode = 1)



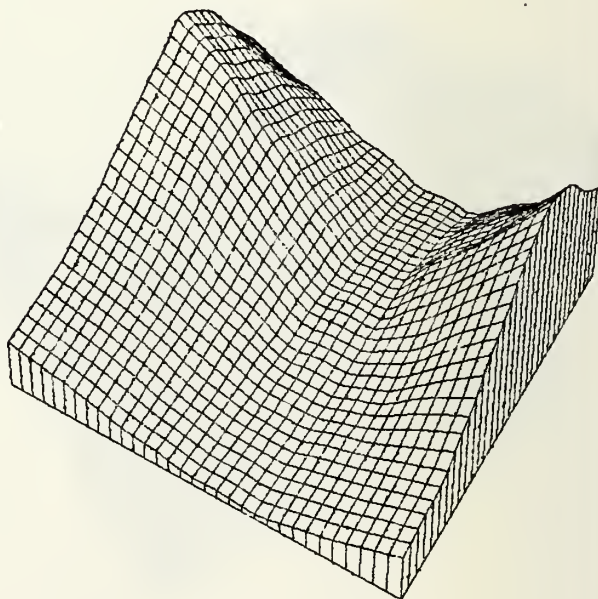
Test Surface
Saddle Function



NPPR = 4 , $E_{\max} = .247$
 $E_{\text{rms}} = .0338$
 $E_{\text{mean}} = .0487$

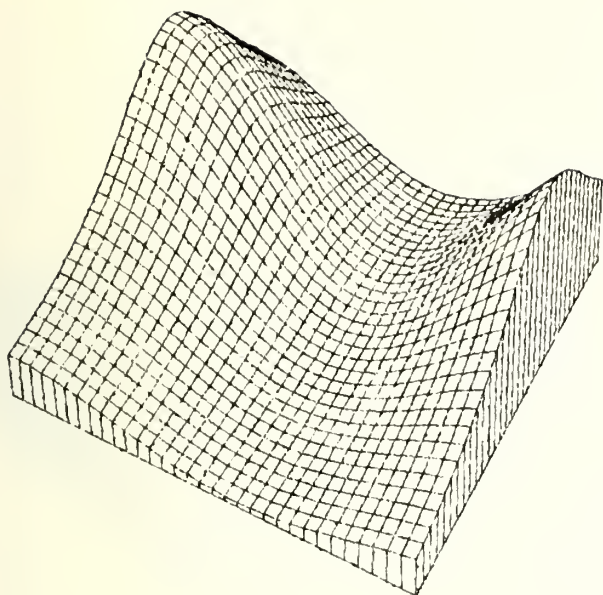


NPPR = 6 , $E_{\max} = .389$
 $E_{\text{rms}} = .0495$
 $E_{\text{mean}} = .0739$

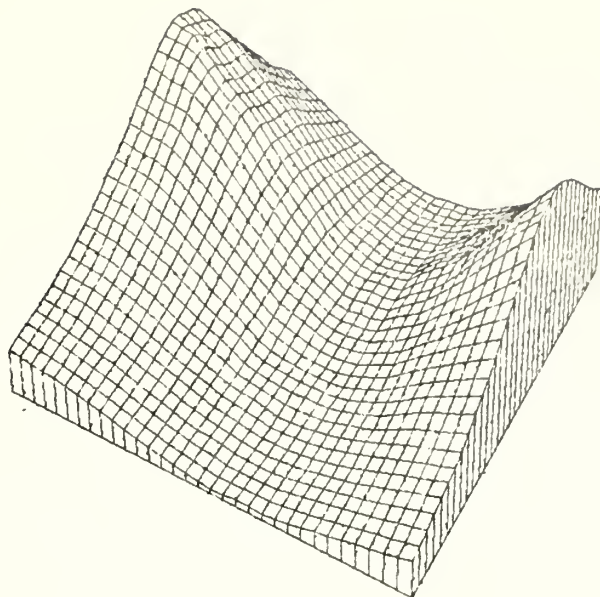


NPPR = 8 , $E_{\max} = .336$
 $E_{\text{rms}} = .0565$
 $E_{\text{mean}} = .0803$

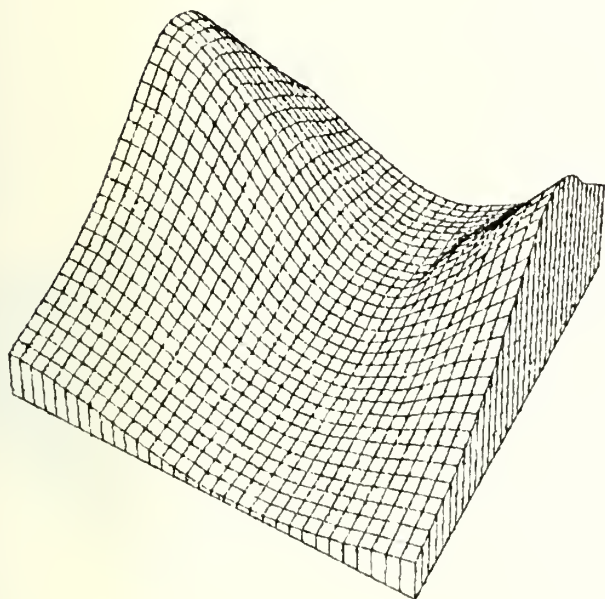
Figure 10 (Mode = 2)



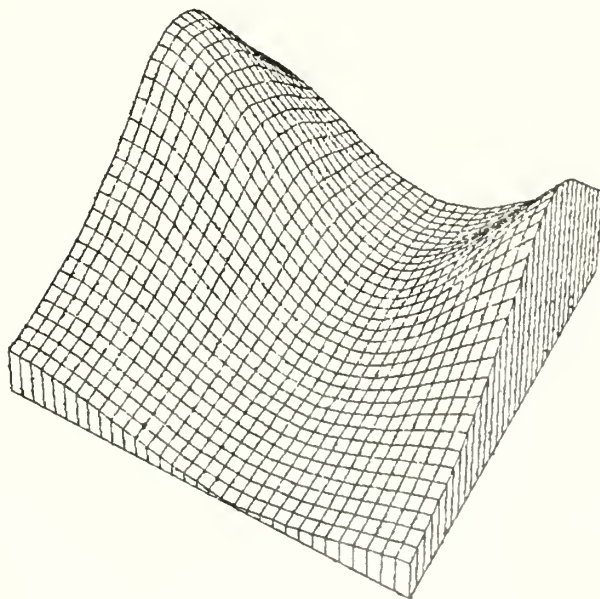
Test Surface
Saddle Function



NPPR = 4 , $E_{\max} = .244$
 $E_{\text{rms}} = .0211$
 $E_{\text{mean}} = .0363$

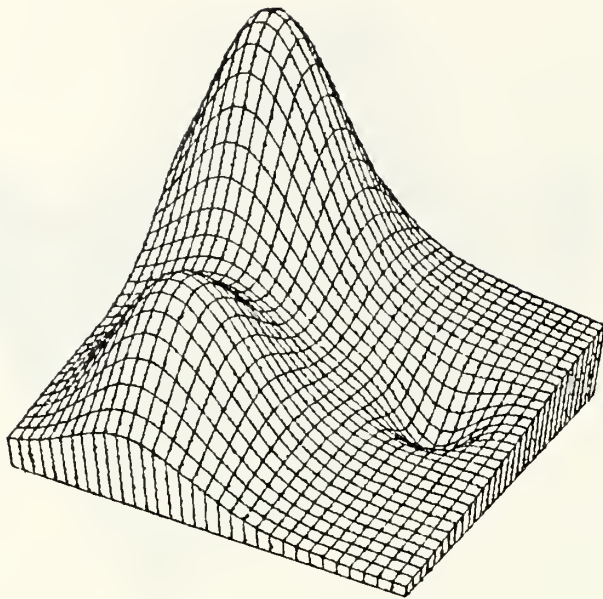


NPPR = 6 , $E_{\max} = .178$
 $E_{\text{rms}} = .0148$
 $E_{\text{mean}} = .0265$

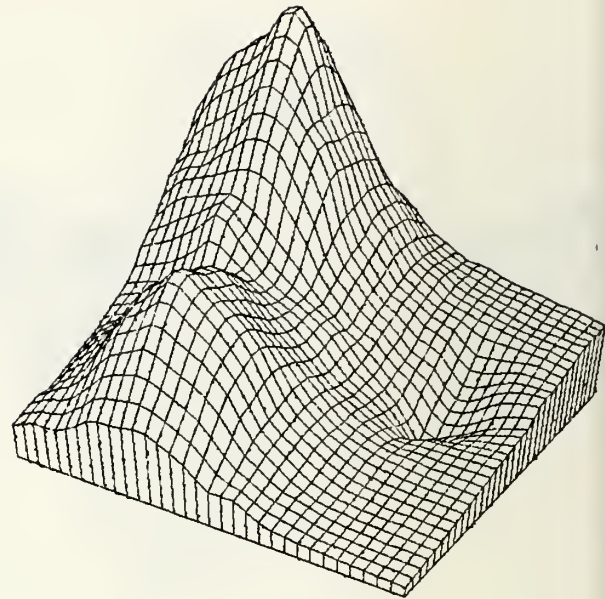


NPPR = 8 , $E_{\max} = .148$
 $E_{\text{rms}} = .0115$
 $E_{\text{mean}} = .0202$

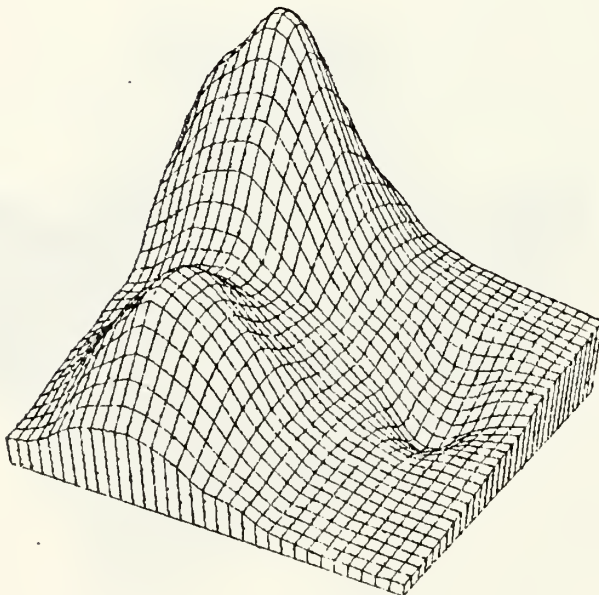
Figure 11 (Mode = 3)



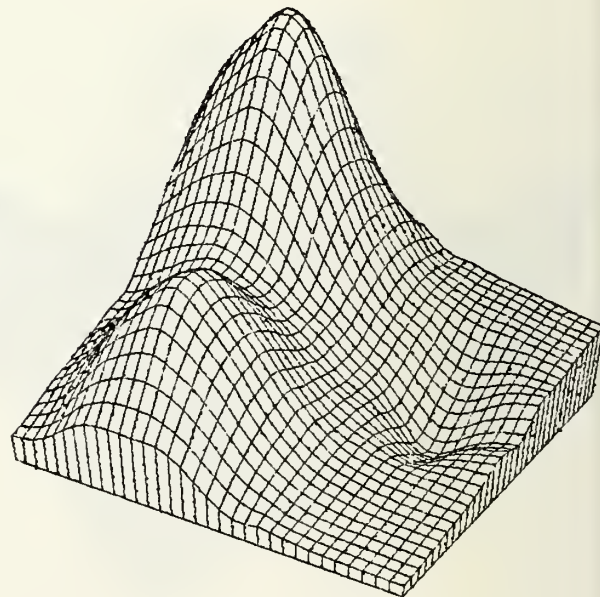
Test Surface
Exponentials



NPPR = 4 , $E_{\max} = 1.20$
 $E_{\text{rms}} = .128$
 $E_{\text{mean}} = .227$

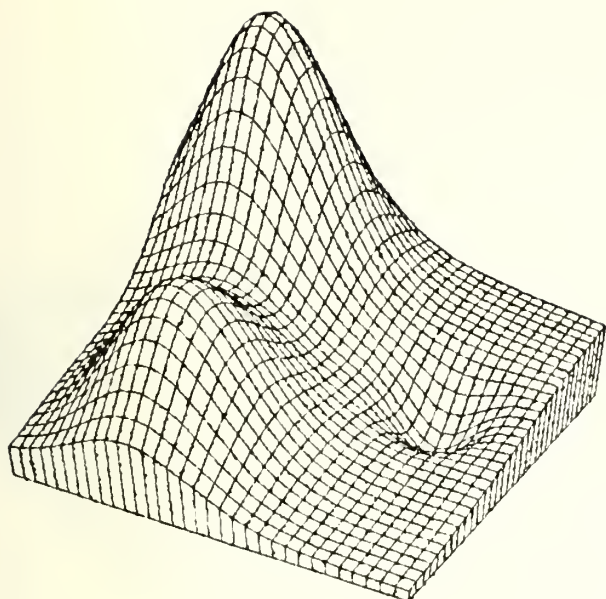


NPPR = 6 , $E_{\max} = .974$
 $E_{\text{rms}} = .0929$
 $E_{\text{mean}} = .169$



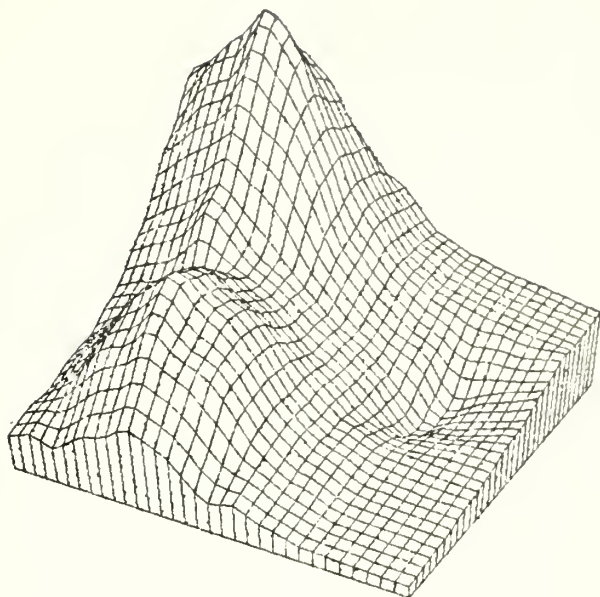
NPPR = 8 , $E_{\max} = .703$
 $E_{\text{rms}} = .0779$
 $E_{\text{mean}} = .129$

Figure 12 (Mode = 1)

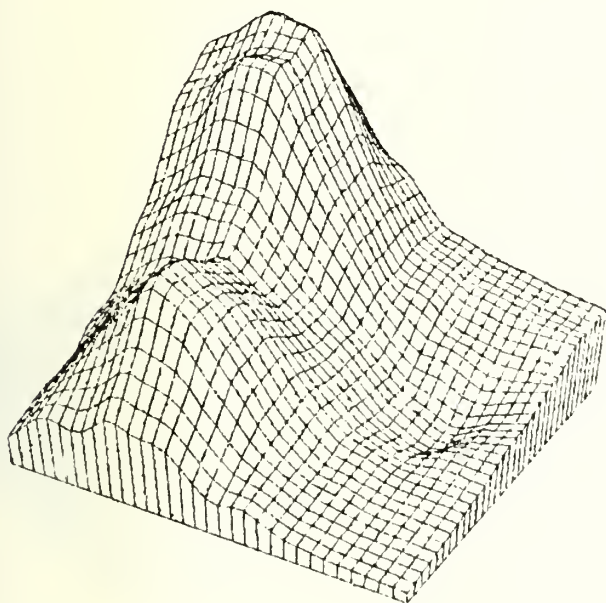


Test Surface

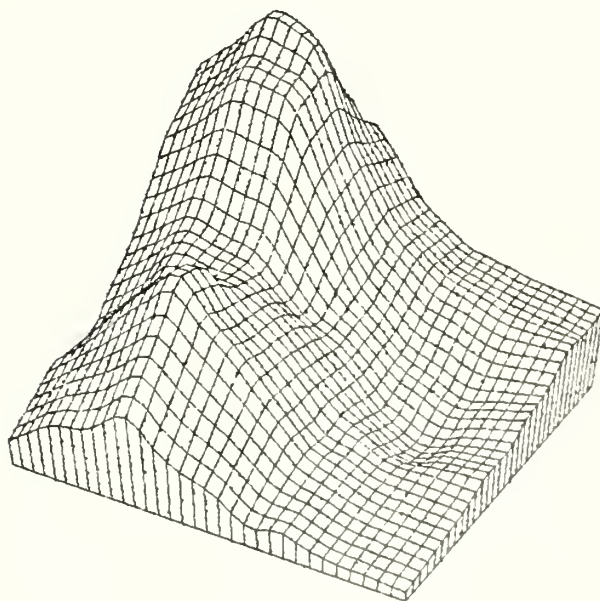
Exponentials



NPPR = 4 , $E_{\max} = 1.29$
 $E_{\text{rms}} = .162$
 $E_{\text{mean}} = .265$

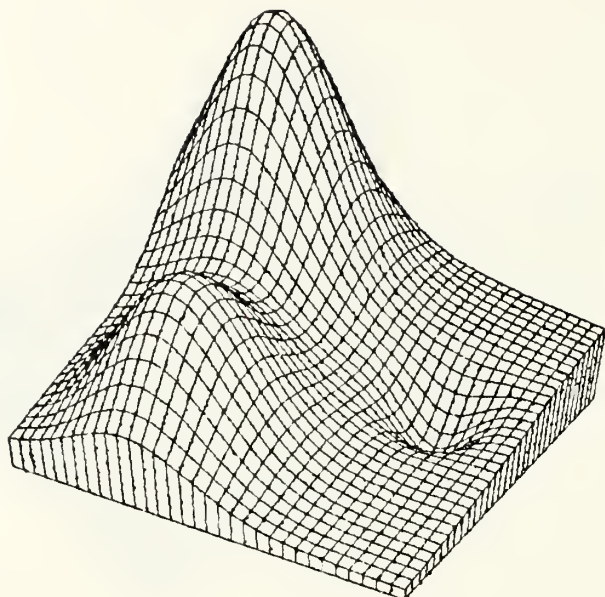


NPPR = 6 , $E_{\max} = 2.16$
 $E_{\text{rms}} = .209$
 $E_{\text{mean}} = .366$



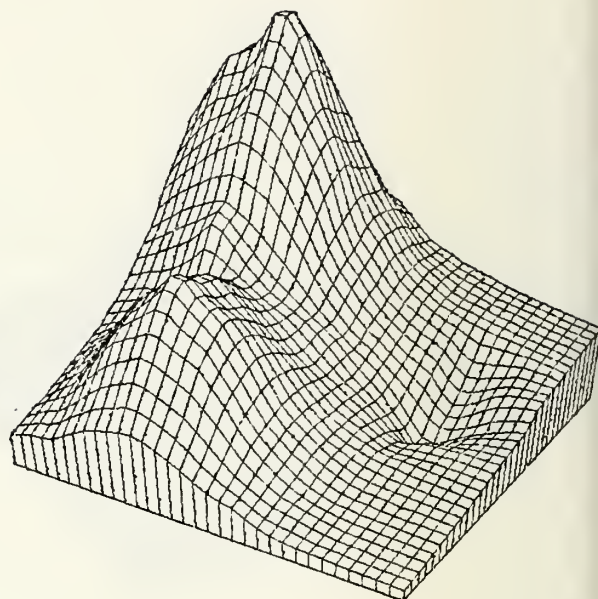
NPPR = 8 , $E_{\max} = 2.04$
 $E_{\text{rms}} = .251$
 $E_{\text{mean}} = .398$

Figure 13 (Mode = 2)

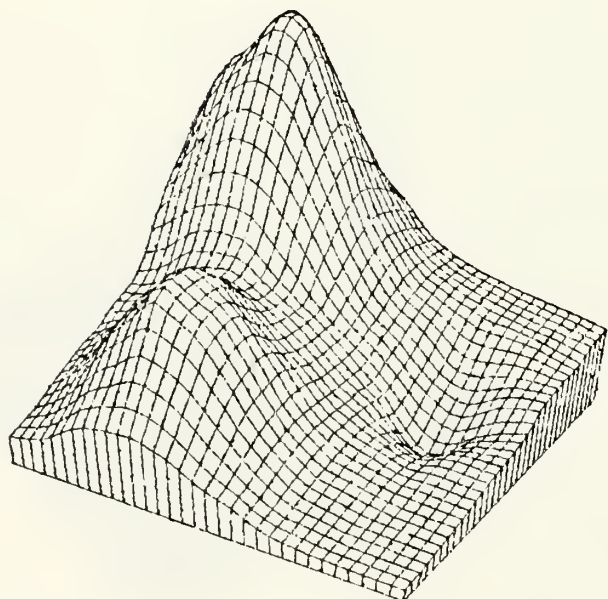


Test Surface

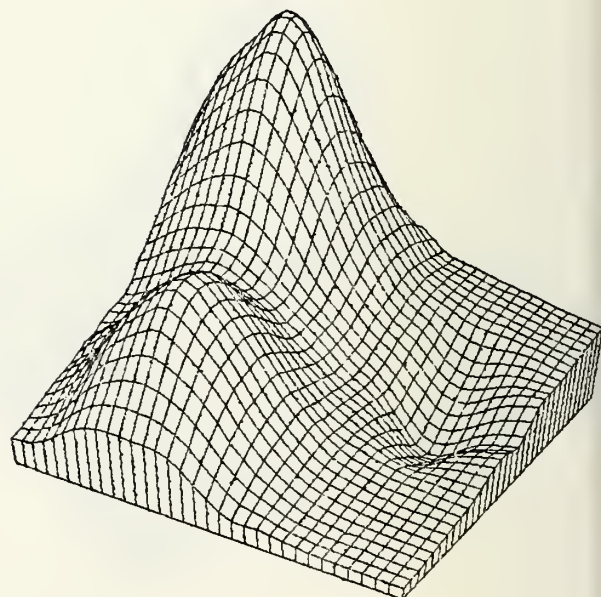
Exponentials



NPPR = 4 , $E_{\max} = 1.30$
 $E_{\text{rms}} = .101$
 $E_{\text{mean}} = .189$



NPPR = 6 , $E_{\max} = .827$
 $E_{\text{rms}} = .0757$
 $E_{\text{mean}} = .133$



NPPR = 3 , $E_{\max} = .748$
 $E_{\text{rms}} = .0778$
 $E_{\text{mean}} = .131$

Figure 14 (Mode = 3)

Appendix: Program Listings and Sample Output

```

DIMENSION X(25), Y(25), F(25), XC(3), YC(3), FD(3,3), FAC(3,3,4),
1 E(3,3), IWK(100), WK(120)
DATA NOUT/6/
DATA FAC / 2.557525042 , 1.156302903 , 0.343660482 ,
1 1.586594904 , 0.729563464 , 0.211667614 ,
2 0.737497040 , 0.343550908 , 0.093714050 ,
3 2.658681228 , 1.522739473 , 0.682157263 ,
4 1.754474583 , 0.877377736 , 0.284611789 ,
5 0.933759918 , 0.413208351 , 0.109657470 ,
6 2.572645288 , 1.158049264 , 0.348002564 ,
7 1.694989367 , 0.724123412 , 0.214020225 ,
8 0.736691672 , 0.341271688 , 0.093442275 ,
9 2.376248312 , 1.107179709 , 0.348220775 ,
A 1.626663927 , 0.747442886 , 0.217782524 ,
B 0.711956600 , 0.324808684 , 0.097759180 /

K = 0
C DO 100 I=1,5
DO 100 J=1,5
K = K+1
X(K) = I-1+1./FLOAT(K)
Y(K) = J-1./FLOAT(K)
100 F(K) = 4.*EXP(-(X(K)**2+Y(K)**2)*.2)

C NP = 25
C DO 120 I=1,3
XO(I) = I+.5
120 YO(I) = I-.5

C DO 160 MODE=1,3
NWK = 120
NIWK = 100
CALL LOB22 (MODE,6,NP,X,Y,F,3,XO,3,YO,IWK,NIWK,WK,NWK,FO,KER)

C DO 140 I=1,3
DO 140 J=1,3
140 E(I,J) = FAC(I,J,MODE)-FO(I,J)

C WRITE (NOUT,2) MODE,KER,NIWK,NWK
160 WRITE (NOUT,1) FO,E

C IWK(1) = 2
IWK(2) = 3
WK(1) = .2
WK(2) = 1.5
WK(3) = 3.5
WK(4) = 4.+1./21.

```



```

WK(5) = 0.25
WK(6) = 1.25
WK(7) = 2.9
WK(8) = 3.8
WK(9) = 4.96
NWK = 120
NIWK = 100
MODE = 3
CALL LOB22 (MODE, 0, NP, X, Y, F, 3, XO, 3, YO, IWK, NIWK, WK, NWK, FO, KER)

C
DO 180 I=1,3
  DO 180 J=1,3
    180 E(I,J) = FO(I,J) - FAC(I,J,4)
C
WRITE (NOUT,2) MODE, KER, NIWK, NWK
WRITE (NOUT,1) FO, E
STOP

C
1 FORMAT (/7X,15HFUNCTION VALUES,3(/3F20.6))//7X,50HDEVIATIONS (THESE
1 VALUES REPRESENT ROUND OFF ERROR)/(3F20.6))
2 FORMAT (/43H)THE VALUES OF MODE, KER, NIWK, AND NWK ARE,4I5)
END
TST00490
TST00500
TST00510
TST00520
TST00530
TST00540
TST00550
TST00560
TST00570
TST00580
TST00590
TST00600
TST00610
TST00620
TST00630
TST00640
TST00650
TST00660
TST00670
TST00680
TST00690
TST00700

```

```

1 SUBROUTINE LOB22 (MODE,NPPR,NPI,XI,YI,FI,NXC,XO,NYO,YO,
      IWK,NIWK,WK,NWK,FO,KER)
      THIS SUBROUTINE SERVES AS A USER INTERFACE TO THE SET OF
      SUBROUTINES THAT IMPLEMENT FRANKEL'S METHOD OF SURFACE INTERPO-
      LATION. RECTANGULAR REGIONS ARE USED WITH PRODUCT QUANTIC
      HERMITE WEIGHT FUNCTIONS. THE RECTANGLES ARE CHOSEN IN AN ATTEMPT
      TO OBTAIN ABOUT NPPR POINTS IN EACH REGION. THE SAME NUMBER OF
      GRID LINES IS USED IN EACH DIRECTION. LOCAL INTERPOLATION FUNC-
      TIONS ARE EITHER OPTIMAL APPROXIMATIONS IN 2,2 OR OPTIMAL APPROX-
      2,2 OR OPTIMAL APPROXIMATIONS IN 2,2 CORNER SPACE SUM THE
      LEAST SQUARES PLANE, OR FOR APPROXIMATION RATHER THAN INTERPO-
      LATION, THE LEAST SQUARES PLANE.
      THE ARGUMENTS ARE AS FOLLOWS.
      MODE - INPUT.
              = 1,
              = 2,
              = 3,
              = 4,
      NPPR - INPUT.
      NPI - INPUT.
      XI - INPUT.
      YI - INPUT.
      FI - INPUT.
      NXO - INPUT.

```

```

10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190
200
210
220
230
240
250
260
270
280
290
300
310
320
330
340
350
360
370
380
390
400
410
420
430
440
450
460
470
480

```

CC

```

XO      - INPUT.      THE VALUES OF X AT WHICH THE INTERPOLATION
500      FUNCTION IS TO BE CALCULATED.
NYO     - INPUT.      THE NUMBER OF YC VALUES AT WHICH THE INTERP-
510      OLATION FUNCTION IS TO BE CALCULATED.
YO      - INPUT.      THE VALUES OF Y AT WHICH THE INTERPOLATION
520      FUNCTION IS TO BE CALCULATED.
IWK     - INPUT AND OUTPUT. THIS ARRAY IS CUTPUT WHEN MODE = 1, 2
530      OR 3, AND IS INPUT WHEN MODE = 4. THIS MUST BE
540      AN ARRAY DIMENSIONED APPROXIMATELY 4*NPI*
550      (1 + 1/NPPR). FOR NPPR = 6, THIS MEANS ABOUT
560      5*NPI. WHEN NPPR IS INPUT AS ZERO THE USER MUST
570      SPECIFY THE NUMBER OF VERTICAL GRID LINES (THE
580      NUMBER OF X TILDA VALUES) IN IWK(1) AND THE
590      NUMBER OF HORIZONTAL GRID LINES (THE NUMBER OF
600      Y TILDA VALUES) IN IWK(2).
610      NIWK - INPUT AND OUTPUT. WHEN MODE = 1, 2, OR 3 THIS MUST BE
620      SET TO THE NUMBER OF LOCAL TICS RESERVED FOR THE
630      ARRAY IWK. ON OUTPUT IT IS SET TO THE ACTUAL
640      REQUIREMENTS FOR THE ARRAY. IF THE ARRAY IWK
650      IS NOT DIMENSIONED LARGE ENOUGH IT MAY CAUSE
660      A FAILURE SINCE THE ARRAY BOUNDS WILL BE
670      EXCEEDED IN SUBROUTINE LOCLIP.
680      WK   - INPUT AND OUTPUT. THIS ARRAY IS CUTPUT WHEN MODE = 1, 2
690      OR 3, AND IS INPUT WHEN MCDE = 4. THIS MUST BE
700      AN ARRAY DIMENSIONED APPROXIMATELY
710      FOR MODE=1, 4*((NPI+SQRT(NPI/NPPR))).
720      FOR MODE=2, 4*((3*NPI/NPPR+SQRT(NPI/NPPR))).
730      FOR MODE=3, 4*((NPI+3*NPI/NPPR+SQRT(NPI/NPPR))).
740      FOR NPPR = 6, THIS MEANS APPROXIMATELY
750      FOR MODE=1, 4*NPI + 1.6*SQRT(NPI)
760      FOR MODE=2, 2*NPI + 1.6*SQRT(NPI)
770      FOR MODE=3, 6*NPI + 1.6*SQRT(NPI)
780      WHEN NPPR IS INPUT AS ZERO THE USER MUST SPECIFY
790      THE VALUES OF X TILDA AND Y TILDA AS FOLLOWS.
800      WK(1) IS MIN(XI), WK(2), ..., WK(NXG+1) ARE THE
810      NXG (= IWK(1)) VALUES OF X TILDA IN INCREASING
820      ORDER, AND WK(NXG+2) IS MAX(XI). WK(NXG+3) IS
830      MIN(YI), WK(NXG+4), ..., WK(NXG+NYG+3) ARE THE
840      (= IWK(2)) VALUES OF Y TILDA IN INCREASING ORDER.
850      AND WK(NXG+NYG+4) IS MAX(YI).
860      NWK  - INPUT AND OUTPUT. ON ENTRY WITH MODE = 1, 2, OR 3, THIS
870      MUST BE THE NUMBER OF LOCATIONS RESERVED FOR THE
880      ARRAY WK. ON OUTPUT IT IS SET TO THE ACTUAL
890      REQUIREMENTS FOR THE ARRAY.
900      FO   - OUTPUT.  VALUES OF THE INTERPOLATION FUNCTION AT THE
910      GRID OF POINTS INDICATED BY NXO, XO, NYC, YC.
920      FO IS ASSUMED TO BE DIMENSIONED (NXO,NYO) IN THE
930      CALLING PROGRAM.
940
950
960

```

```

CCCCCCCCCCCCCCCC
KER  -  OUTPUT.
      = 0,      RETURN INDICATOR.
      = -1,     NORMAL RETURN.
      = 1,     PROBLEM HAS NOT BEEN PREVIOUSLY SET UP (LOB22
      = 2,     CALLED WITH MODE = 1, 2 OR 3)
      = 3,     ERROR RETURN FROM CLOB22, SINGULAR MATRIX IN THE
      = 4,     LEAST SQUARES FIT.
      = 5,     ERROR RETURN FROM CLOB22, SINGULAR MATRIX IN THE
      = 6,     OPTIMAL FIT.
          ERROR RETURN FROM CLOB22. SOME RECTANGLE (I,J)
          HAS MORE THAN THE ALLOWED NUMBER OF POINTS
          ASSOCIATED WITH IT.
          SUBROUTINE CLOB22.
          PREVIOUS ERROR RETURN FROM CLOB22 HAS NOT BEEN
          CORRECTED.
          IWK AND WK ARRAYS HAVE NOT BEEN DIMENSIONED
          LARGE ENOUGH IN THE CALLING PROGRAM. REDIMEN-
          SION IWK AND WK TO AT LEAST THE SIZE INDICATED
          BY NIWK AND NWK, RESPECTIVELY.
          MODE IS OUT OF RANGE.

          DIMENSION XI(1), YI(1), FI(1), IWK(1), WK(1), XO(1), YO(1), FC(NXO
1,1)
DATA KERO/-1/
IF (MODE.LT.1.OR.MODE.GT.4) GO TO 220
KER = 0

ON INITIAL ENTRY MODE = 1, 2, OR 3, AND THE GRID LINES ARE SET UP,
LOCAL INTERPOLATION POINTS ARE DETERMINED AND LOCAL APPROXIMATIONS
ARE COMPUTED.

IF (MODE.EQ.4) GO TO 140
NWKIN = NWK
NIWKIN = NIWK
NXGKW = 1
NPWK = 3
IF (NPPR.LE.0) GO TO 100
NXG = SQRT(4.*FLOAT(NPI)/FLOAT(NPPR))-5
NYG = NXG
IWK(1) = NXG
IWK(2) = NYG
GC TO 120
100 NXG = IWK(1)
120 NYG = IWK(2)
IALWK = NXG+NYG+5
IABWK = IALWK
IF (MODE.NE.1) IABWK = IABWK+3*NXG*NYG
NYGKW = NXG+3
MPWK = NXG*NYG+4

```



```

C      IF (NPPR.GT.0) CALL GRID (XI,YI,NPI,NXG,WK(NXGWK),NYG,WK(NYGWK),WKLOB 1450
C      1(IALWK)) 1460
C      DETERMINE THE LOCAL INTERPOLATION POINTS FOR THE REGIONS. 1470
C      CALL LOCLIP (NXG,WK(NXGWK),NYG,WK(NYGWK),NPI,XI,YI,IWK(NPWK),IWK(MLOB 1480
C      1PWK),WK(IALWK)) 1490
C      NWK = IABWK-1 1500
C      IF (MODE.NE.2) NWK = NWK+IWK(MPWK-1)-1 1510
C      NIWK = NXG*NYG+2+IWK(MPWK-1) 1520
C      IF (NIWK.GT.NIWKIN) GO TO 200 1530
C      IF (NWK.GT.NWKIN) GO TO 200 1540
C      MC = MODE 1550
C      1560
C      1570
C      1580
C      1590
C      1600
C      CALL CLOB22 (MO,XI,YI,FI,NXG,WK(NXGWK),NYG,WK(NYGWK),IWK(NPWK),IWKLOB 1610
C      1(MPWK),WK(IALWK),WK(IABWK),IER) 1620
C      KERO = IER 1630
C      IF (IER.NE.0) GO TO 160 1640
C      140 IF (KERO.NE.0) GO TO 180 1650
C      1660
C      1670
C      1680
C      1690
C      CALL EVLB22 (MO,XI,YI,NXG,WK(NXGWK),NYG,WK(NYGWK),IWK(NPWK),IWK(MPLOB 1700
C      1WK),WK(IALWK),WK(IABWK),NXO,XO,NYC,YO,FO) 1710
C      RETURN 1720
C      1730
C      1740
C      1750
C      1760
C      1770
C      1780
C      1790
C      1800
C      1810
C      1820
C      1830
C      1840

```



```

490
500
510
520
530
540
550
560
570
580
590
600
610
GRI
GRI
GRI
GRI
GRI
GRI
GRI
GRI
GRI
GRI
GRI

```

```

C      IPL = I+1
      DO 240 J=IPL,N
      IF (T(I).LE.T(J)) GO TO 240
      TS = T(I)
      T(I) = T(J)
      T(J) = TS
      240 CONTINUE
C      260 CONTINUE
C      GO TO (120,180), K
      END

```

```

SUBROUTINE LOCLIP (NXG,XG,NYG,YG,NPI,XI,YI,NP,MP,D)
THIS SUBROUTINE DETERMINES THE LCCAL INTERPCLATION POINTS FOR THE
GRID VERSION OF FRANK'S METHOD OF SURFACE INTERPOLATION.
MINPTS POINTS ARE REQUIRED FOR EACH REGION.
IF FEWER THAN MINPTS POINTS ARE FOUND IN THE REGION, THE NEXT
CLOSEST POINTS (IN THE SUP NORM AFTER THE CURRENT RECTANGLE IS
TRANSFORMED ONTO (0,1)) ARE USED. MINPTS IS SET TO 3, WHICH IS
THE RECOMMENDED VALUE, ALTHOUGH IT MAY BE ALTERED.

THE ARGUMENTS ARE AS FOLLOWS.

      NXG - INPUT:  NUMBER OF VERTICAL GRID LINES.
      XG  - INPUT:  THE COORDINATES OF THE VERTICAL GRID LINES.
      NYG - INPUT:  NUMBER OF HORIZONTAL GRID LINES.
      YG  - INPUT:  THE COORDINATES OF THE HORIZONTAL GRID LINES.
      NPI - INPUT:  THE NUMBER OF DATA POINTS.
      XI  - INPUT:  THE DATA POINTS (XI,YI,FI), I=1,...,NPI.
      YI  - INPUT:
      FI  - INPUT:
      NP  - OUTPUT. AN ARRAY WHICH GIVES THE INITIAL SUBSCRIPT IN
      THE ARRAY MP AT WHICH THE SUBSCRIPTS FOR THE
      LOCAL INTERPOLATION POINTS ARE STORED.
      MP  - OUTPUT. AN ARRAY WHICH GIVES THE SUBSCRIPTS FOR THE
      LOCAL INTERPOLATION POINTS.
      D   - A WORK ARRAY OF DIMENSION AT LEAST NPI.

      DIMENSION XG(1), YG(1), XI(1), YI(1), NP(1), MP(1), D(1)
      DATA MINPTS/3/
      IJ = 1
      NP(1) = 1
      L = 0

      DO 200 J=1,NYG
      YGA = (YG(J+2)+YG(J))/2.
      DYG = YG(J+2)-YG(J)

      DO 180 I=1,NXG
      XGA = (XG(I+2)+XG(I))/2.
      DXG = XG(I+2)-XG(I)
      IJ = IJ+1

      DETERMINE THE POINTS IN THE (I,J)TH RECTANGLE.

      DC 120 NK=1,NPI
      IF (XI(NK).GT.XG(I+2).OR.XI(NK).LT.XG(I)) GC TO 100
      IF (YI(NK).GT.YG(J+2).OR.YI(NK).LT.YG(J)) GC TO 100

```

```

      L = L+1
      D(NK) = 1.E10
      MP(L) = NK
      GO TO 120
100 D(NK) = AMAX1(ABS(XI(NK)-XGA)/DXG,ABS(YI(NK)-YGA)/DYG)
120 CCNTINUE
C
      NP(IJ) = L+1
      IF (NP(IJ)-NP(IJ-1).GE.MINPTS) GO TO 180
C
      ADD THE CLOSEST PCINTS IF THERE ARE LESS THAN MINPTS IN THE
      RECTANGLE.
C
      LV = MINPTS-(NP(IJ)-NP(IJ-1))
C
      DC 160 II=1,LM
      L = L+1
      MP(L) = 1
      DM = D(1)
C
      DO 140 NK=2,NPI
      IF (D(NK).GE.DM) GO TO 140
      DM = D(NK)
      MP(L) = NK
      CONTINUE
140
C
      NK = MP(L)
      D(NK) = 1.E10
C
      NP(IJ) = L+1
      CONTINUE
180
C
      CONTINUE
200
      RETURN
      END

```

```

LOC 490
LOC 500
LOC 510
LOC 520
LOC 530
LOC 540
LOC 550
LOC 560
LOC 570
LOC 580
LOC 590
LOC 600
LOC 610
LOC 620
LOC 630
LOC 640
LOC 650
LOC 660
LOC 670
LOC 680
LOC 690
LOC 700
LOC 710
LOC 720
LOC 730
LOC 740
LOC 750
LOC 760
LOC 770
LOC 780
LOC 790
LOC 800
LOC 810
LOC 820
LOC 830
LOC 840

```


[illegible]

```

C      CALL LEQTP (C,1,3,C(7),1,0,D1,D2,KER)
C      IF (KER.NE.0) GO TO 280
C      DO 140 LL=1,3
C      IAL=IALS+LL
C      140 AL(IAL) = C(LL+6)
C      160 IF (MODL.EQ.2) GO TO 240
C      CALCULATE THE B CORNER 2,2 OPTIMAL APPROXIMATION
C      KK = 0
C      DO 200 LI=1,LEND
C      MPI = NP(IJ)+LI-1
C      KI = MP(MPI)
C      XKI = (XI(KI)-XG(I))/DX
C      YKI = (YI(KI)-YG(J))/DY
C      DO 180 LJ=1,LJ-1
C      MPJ = NP(IJ)+LJ-1
C      KJ = MP(MPJ)
C      XKJ = (XI(KJ)-XG(I))/DX
C      YKJ = (YI(KJ)-YG(J))/DY
C      KK = KK+1
C      180 C(KK) = K(A,B,XKI,YKI,XKJ,YKJ)
C      LB = LBS+LI
C      C(LB) = FI(KI)
C      IF (MODL.EQ.1) GO TO 200
C      XIK = XI(KI)-XG(I+1)
C      YIK = YI(KI)-YG(J+1)
C      C(LB) = C(LB)+(AL(IALS+1)+AL(IALS+2)*XIK+AL(IALS+3)*YIK)
C      200 CONTINUE
C      CALL LEQTP (C,1,LEND,C(LBS+1),1,0,D1,D2,KER)
C      IF (KER.NE.0) GO TO 300
C      DO 220 LI=1,LEND
C      IAB = NP(IJ)+LI-1
C      LB = LBS+LI
C      220 AB(IAB) = C(LB)
C      A = 0.
C      240 CONTINUE
C      260 B = 0.

```

```

CLO 970
CLO 980
CLO 990
CLO 1000
CLO 1010
CLO 1020
CLO 1030
CLO 1040
CLO 1050
CLO 1060
CLO 1070
CLO 1080
CLO 1090
CLO 1100
CLO 1110
CLO 1120
CLO 1130
CLO 1140
CLO 1150
CLO 1160
CLO 1170
CLO 1180
CLO 1190
CLO 1200
CLO 1210
CLO 1220
CLO 1230
CLO 1240
CLO 1250
CLO 1260
CLO 1270
CLO 1280
CLO 1290
CLO 1300
CLO 1310
CLO 1320
CLO 1330
CLO 1340
CLO 1350
CLO 1360
CLO 1370
CLO 1380
CLO 1390
CLO 1400
CLO 1410
CLO 1420
CLO 1430
CLO 1440

```

```

C      RETURN
C
C      ERROR RETURNS
C
C      280 IER = 1
C          GC TO 320
C      300 IER = 2
C      320 WRITE (NOUT,1) I,J,IER
C          MPST = NP(IJ)
C          MPSE = MPST+LEND-1
C
C      DO 340 MPS=MPST,MPSE
C          I = MP(MPS)
C      340 WRITE (NOUT,2) I,XI(I),YI(I),FI(I)
C
C      RETURN
C      IER = 3
C      360 WRITE (NOUT,3) I,J,LEND,NC
C          RETURN
C
C      1 FORMAT (46HOSINGULAR MATRIX DETECTED FOR GRID POINT I,J =,2I5,12H
C          1, IER = ,12//40H THE DATA POINTS INVOLVE ARE AS FOLLOWS/)
C      2 FORMAT (1X,15,3E15.6)
C      3 FORMAT (15H0THE RECTANGLE, I3,1H,,I3,5H HAS ,I3,27H POINTS ASSOCIACLO
C          1TED WITH IT /51H THE CURRENT LIMIT IMPOSED IN SUBROUTINE CLC822 ISCLC
C          2,I3/84H THIS MAY BE MODIFIED BY REDEFINING NC AND CHANGING THE DICLO
C          3MENSION OF C APPROPRIATELY)
C          END

```



```

1 SUBROUTINE EVLB22 (MODL,XI,YI,NXG,XG,NYG,YG,NP,MP,AL,AB,
    NXO,XO,NYO,YO,FO)
    THIS SUBROUTINE EVALUATES THE INTERPOLANT FOR THE GRID VERSION OF
    FRANKEL'S METHOD. THE FUNCTION IS EVALUATED AT THE GRID OF POINTS
    INDICATED BY NXO, XO, NYO, YO, AND THESE VALUES ARE RETURNED
    IN THE ARRAY FO, WHICH IS ASSUMED TO BE DIMENSIONED (NXO,NYO).

    THE ARGUMENTS ARE AS FOLLOWS.

    MODL - INPUT.
            = 1, SPECIFIES THE TYPE OF LOCAL APPROXIMATION USED.
            = 2, USED THE OPTIMAL APPROXIMATION IN B CORNER 2,2.
            = 3, USED THE LEAST SQUARES PLANE.
            BOCLEAN SUM THE LEAST SQUARES PLANE.

    XI - INPUT. THE DATA POINTS (XI,YI,FI), I=1,...,NPI.
    YI - INPUT. THE NUMBER OF VERTICAL GRID LINES.
    FI - INPUT. THE COORDINATES OF THE VERTICAL GRID LINES.
    NXG - INPUT. THE COORDINATES OF HORIZONTAL GRID LINES.
    XG - INPUT. THE NUMBER OF HORIZONTAL GRID LINES.
    NYG - INPUT. THE COORDINATES OF THE HORIZONTAL GRID LINES.
    YG - INPUT. AN ARRAY WHICH GIVES THE INITIAL SUBSCRIPT IN
    NP - INPUT. THE ARRAY MP AT WHICH THE SUBSCRIPTS FOR THE
    LOCAL INTERPOLATION POINTS ARE STORED.
    MP - INPUT. AN ARRAY WHICH GIVES THE SUBSCRIPTS FOR THE
    LOCAL INTERPOLATION POINTS.
    AL - INPUT. THE COEFFICIENTS FOR THE LEAST SQUARES PLANE,
    AB - INPUT. WHEN MODL = 2 OR 3.
    NXO - INPUT. THE COEFFICIENTS FOR THE OPTIMAL APPROXIMATION
    XO - INPUT. WHEN MODL = 1 OR 3.
    NYO - INPUT. THE NUMBER OF XC VALUES AT WHICH THE INTERPO-
    YO - INPUT. LATION FUNCTION IS TO BE CALCULATED.
    FO - OUTPUT. THE VALUES OF X AT WHICH THE INTERPOLATION
    FUNCTION IS TO BE CALCULATED.
    THE NUMBER OF Y VALUES AT WHICH THE INTERPO-
    LATION FUNCTION IS TO BE CALCULATED.
    THE VALUES OF Y AT WHICH THE INTERPOLATION
    FUNCTION IS TO BE CALCULATED.
    VALUES OF THE INTERPOLATION FUNCTION AT THE
    GRID POINTS INDICATED BY NXO, XO, NYO, YO.
    FO IS ASSUMED TO BE DIMENSIONED (NXO,NYO) IN THE
    CALLING PROGRAM.

    DIMENSION XG(1), YG(1), XI(1), YI(1), NP(1), MP(1), FC(4), AL(1),
    LAB(1), XO(1), YO(1), FO(NXO,1)

    BECAUSE OF THE SHORT WORD LENGTH OF THE IBM 360/370 COMPUTERS

```

```

C      CERTAIN VARIABLE ARE DECLARED AS DOUBLE PRECISION.  THIS STATE-
C      MENT MAY BE SAFELY REPLACED WITH THE STATEMENT 'REAL K' WHEN
C      THIS PROGRAM IS USED ON COMPUTERS WITH LONGER WORD LENGTHS.
C
C      DOUBLE PRECISION K
C
C      ARITHMETIC STATEMENT FUNCTION FOR THE HERMITE QUINTIC.
C
C      H5(S) = 1.-S**3*((6.*S-15.)*S+10.)
C      J = 1
C
C      DC 640 JC=1,NY0
C
C      DETERMINE THE LOCATION OF THE POINT Y0 IN TERMS OF THE SMALLEST
C      VALUE OF J SUCH THAT Y0(J0) IS IN SOME RECTANGLE (I,J).
C
C      YV = Y0(J0)
C      JJS = J+1
C      IF (YV.LT.YG(JJS)) JJS=1
C
C      DO 100 JJ=JJS,NYG
C      IF (YV.LT.YG(JJ+1)) GO TO 120
C      100 CONTINUE
C
C      J = NYG
C      GC TO 140
C      J = JJ-1
C      120
C      140 JD = 3
C      IF (J.GE.1) GO TO 160
C      JD = 0
C      J = 1
C      GO TO 180
C      IF (J.LT.NYG) GO TO 180
C      JC = 6
C      160 DY = YG(J+2)-YG(J+1)
C      180 I = 1
C
C      DO 620 IC=1,NX0
C
C      DETERMINE THE LOCATION OF THE POINT X0 IN TERMS OF THE SMALLEST
C      VALUE OF I SUCH THAT X0(10) IS IN THE RECTANGLE (I,J).
C
C      IIS = I+1
C      XV = X0(10)
C      IF (XV.LT.XG(IIS)) IIS=1
C
C      DC 200 II=IIS,NXG
C      IF (XV.LT.XG(II+1)) GO TO 220

```

```

EVL 490
EVL 500
EVL 510
EVL 520
EVL 530
EVL 540
EVL 550
EVL 560
EVL 570
EVL 580
EVL 590
EVL 600
EVL 610
EVL 620
EVL 630
EVL 640
EVL 650
EVL 660
EVL 670
EVL 680
EVL 690
EVL 700
EVL 710
EVL 720
EVL 730
EVL 740
EVL 750
EVL 760
EVL 770
EVL 780
EVL 790
EVL 800
EVL 810
EVL 820
EVL 830
EVL 840
EVL 850
EVL 860
EVL 870
EVL 880
EVL 890
EVL 900
EVL 910
EVL 920
EVL 930
EVL 940
EVL 950
EVL 960

```

```

C      200 CONTINUE
      I = NXG
      GO TO 240
220    I = I-1
240    ID = 2
      IF (I.GE.1) GO TO 260
      IC = 1
      I = 1
      GO TO 280
260    IF (I.LT.NXG) GO TO 280
      ID = 3
280    DX = XG(I+2)-XG(I+1)
      KC = ID+JD
      A = 0.
      IF (I.EQ.1) A = 1.
      B = 0.
      IF (J.EQ.1) B = 1.
      GC TO (300,360,300,440,520,440,300,360,300), KD
      THIS IS FOR (XO(IO),YO(JO)) POINTS IN A SINGLE RECTANGLE (I,J)
C
C      300 FV = 0.
      IJ = (J-1)*NXG+I
      IAL = 3*IJ-2
      IF (MODL.EQ.2) GO TO 340
      LMAX = NP(IJ+1)-NP(IJ)
      DXA = XG(I+2)-XG(I)
      DYA = YG(J+2)-YG(J)
      XVD = (XV-XG(I))/DXA
      YVD = (YV-YG(J))/DYA
C
      DC 320 L=1,LMAX
      MPS = NP(IJ)+L-1
      KI = MP(MPS)
      XKI = (XI(KI)-XG(I))/DXA
      YKI = (YI(KI)-YG(J))/DYA
      320 FV = FV+AB(MPS)*K(A,B,XKI,YKI,XVD,YVD)
C
      340 IF (MODL.NE.1) FV = FV+AL(IAL)+AL(IAL+1)*(XV-XG(I+1))+AL(IAL+2)*(YV-YG(J+1))
      GO TO 620
C
      THIS IS FOR XO(IO),YO(JO)) POINTS WHICH ARE IN TWO RECTANGLES,
      (I,J) AND (I+1,J).
C
C      360 DYA = YG(J+2)-YG(J)
      YVD = (YV-YG(J))/DYA

```

```

EVL 970
EVL 980
EVL 990
EVL 1000
EVL 1010
EVL 1020
EVL 1030
EVL 1040
EVL 1050
EVL 1060
EVL 1070
EVL 1080
EVL 1090
EVL 1100
EVL 1110
EVL 1120
EVL 1130
EVL 1140
EVL 1150
EVL 1160
EVL 1170
EVL 1180
EVL 1190
EVL 1200
EVL 1210
EVL 1220
EVL 1230
EVL 1240
EVL 1250
EVL 1260
EVL 1270
EVL 1280
EVL 1290
EVL 1300
EVL 1310
EVL 1320
EVL 1330
EVL 1340
EVL 1350
EVL 1360
EVL 1370
EVL 1380
EVL 1390
EVL 1400
EVL 1410
EVL 1420
EVL 1430
EVL 1440

```

```

C      DC 420 IP=1,2
      FC(IP) = 0.
      IS = I+IP-1
      IJ = (J-1)*NXG+IS
      IAL = 3*IJ-2
      IF (MODL.EQ.2) GO TO 400
      DXA = XG(IS+2)-XG(IS)
      XVD = (XV-XG(IS))/DXA
      LMAX = NP(IJ+1)-NP(IJ)

C      DO 380 L=1,LMAX
      MPS = NP(IJ)+L-1
      KI = MP(MPS)
      XKI = (XI(KI)-XG(IS))/DXA
      YKI = (YI(KI)-YG(J))/DYA
      FC(IP) = FC(IP)+AB(MPS)*K(A,B,XKI,YKI,XVD,YVD)
380   FC(IP) = FC(IP)=FC(IP)+AL(IAL)+AL(IAL+1)*(XV-XG(IS+1))+AL(IAL+1)
C      1L+2)*(YV-YG(J+1))
      A=0.
      420 CONTINUE

C      WI = H5((XV-XG(I+1))/DX)
      FV = FC(1)*WI+(1.-WI)*FC(2)
      GC TO 620

C      T+IS IS FOR (XO(IC),YO(JO)) POINTS WHICH ARE IN TWO RECTANGLES,
C      (I,J) AND (I,J+1).
C      440 DXA = XG(I+2)-XG(I)
      XVD = (XV-XG(I))/DXA

C      DC 500 JP=1,2
      FC(JP) = 0.
      JS = J+JP-1
      IJ = (JS-1)*NXG+I
      IAL = 3*IJ-2
      IF (MODL.EQ.2) GO TO 480
      DYA = YG(JS+2)-YG(JS)
      YVD = (YV-YG(JS))/DYA
      LMAX = NP(IJ+1)-NP(IJ)

C      DO 460 L=1,LMAX
      MPS = NP(IJ)+L-1
      KJ = MP(MPS)
      XKJ = (XI(KJ)-XG(I))/DXA
      YKJ = (YI(KJ)-YG(JS))/DYA

```



```

C 460 FC(JP) = FC(JP)+AB(MPS)*K(A,B,XKJ,YKJ,XVD,YVD)
C 480 IF (MODL.NE.1) FC(JP)=FC(JP)+AL(IAL)+AL(IAL+1)*(XV-XG(I+1))+AL(IAL+2)*(YV-YG(JS+1))
      B = 0.
500 CONTINUE
C
      UJ = H5((YV-YG(J+1))/DY)
      FV = FC(1)*UJ+(1.-UJ)*FC(2)
      GC TO 620
C
      THIS IS FOR (XO(IC),YO(JO)) POINTS WHICH ARE IN FOUR RECTANGLES,
      (I,J), (I+1,J), (I,J+1), AND (I+1,J+1).
C
520 KFC = 3
C
      DQ 600 JP=1,2
      JS = J+JP-1
      DYA = YG(JS+2)-YG(JS)
      YVD = (YV-YG(JS))/DYA
      A = 1.
C
      DQ 580 IP=1,2
      IS = I+IP-1
      IJ = (JS-1)*NXG+IS
      IAL = 3*IJ-2
      KFC = KFC+1
      FC(KFC) = 0.
      IF (MODL.EQ.2) GO TO 56J
      IF (IS.GT.1) A = 0.
      DXA = XG(IS+2)-XG(IS)
      XVD = (XV-XG(IS))/DXA
      LMAX = NP(IJ+1)-NP(IJ)
C
      DC 540 L=1,LMAX
      MPS = NP(IJ)+L-1
      KI = MP(MPS)
      XKI = (XI(KI)-XG(IS))/DXA
      YKI = (YI(KI)-YG(JS))/DYA
540 FC(KFC) = FC(KFC)+AB(MPS)*K(A,B,XKI,YKI,XVD,YVD)
C
560 IF (MODL.NE.1) FC(KFC)=FC(KFC)+AL(IAL)+AL(IAL+1)*(XV-XG(IS+1))+AL(IAL+2)*(YV-YG(JS+1))
      LIAL+2)
580 CCNTINUE
C
      B = 0.
600 CONTINUE
C

```

EVL 193J
 EVL 1940
 EVL 1950
 EVL 1960
 EVL 1970
 EVL 1980
 EVL 1990
 EVL 2000
 EVL 2010
 EVL 2020
 EVL 2030
 EVL 2040
 EVL 2050
 EVL 2060
 EVL 2070
 EVL 2080
 EVL 2090
 EVL 2100
 EVL 2110
 EVL 2120
 EVL 2130
 EVL 2140
 EVL 2150
 EVL 2160
 EVL 2170
 EVL 2180
 EVL 2190
 EVL 2200
 EVL 2210
 EVL 2220
 EVL 2230
 EVL 2240
 EVL 2250
 EVL 2260
 EVL 2270
 EVL 2280
 EVL 2290
 EVL 2300
 EVL 2310
 EVL 2320
 EVL 2330
 EVL 2340
 EVL 2350
 EVL 2360
 EVL 2370
 EVL 2380
 EVL 2390
 EVL 2400

WI = H5((XV-XG(I+1))/DX)	EVL	2410
UJ = H5((YV-YG(J+1))/DY)	EVL	2420
FV = WI*(UJ*FC(1)+(1.-UJ)*FC(3))+(1.-WI)*(UJ*FC(2)+(1.-UJ)*FC(4))	EVL	2430
62C F0(I0,J0) = FV	EVL	2440
C	EVL	2450
64C CCNTINUE	EVL	2460
C	EVL	2470
RETURN	EVL	2480
END	EVL	2490

```

10 FUNCTION K (A,B,U,V,S,T)
20 THIS FUNCTION EVALUATES THE REPRESENTER FOR THE PCINT EVALUATION
30 (AT (U,V)) FUNCTIONAL FOR THE SARD CORNER SPACE B CORNER 2,2.
40
50 THE ARGUMENTS ARE AS FOLLOWS.
60
70     A,B - INPUT.  THE BASE POINT (A,B) FOR THE SARD SPACE B
80     CORNER 2,2.
90     U,V - INPUT.  THE LINEAR FUNCTIONAL IS EVALUATION AT (U,V).
100    S,T - INPUT.  THE REPRESENTER IS EVALUATED AT (S,T).
110
120 BECAUSE OF THE SHCRT WORD LENGTH CF THE IBM 360/370 COMPUTERS
130 CERTAIN VARIABLES ARE DECLARED AS DOUBLE PRECISION. THIS STATE-
140 MENT MAY BE SAFELY REPLACED WITH THE STATEMENT 'REAL K', WHEN
150 THIS PROGRAM IS USED CN COMPUTERS WITH LONGER WORD LENGTHS.
160
170 DCUBLE PRECISION K,SMA,UMA,USMA,X,TRMS,GP1,GP2
180 SS = S
190 UU = U
200 AA = A
210 KVAR = 1
220 SMA = SS-AA
230 UMA = UU-AA
240 USMA = SMA*UMA
250 TRMS = 0.
260 IF (USMA.LE.0.) GO TO 160
270 IF (UMA.GE.0.) GO TO 120
280 UMA = -UMA
290 SMA = -SMA
300 IF (SMA.LE.UMA) GO TO 140
310 X = SMA
320 SMA = UMA
330 UMA = X
340 TRMS = SMA/2.*(USMA-SMA**2/3.)
350 GP2 = 1.+USMA+TRMS
360 GO TO (180,200), KVAR
370 GP1 = GP2
380 SS = T
390 UU = V
400 AA = B
410 KVAR = 2
420 GC TO 100
430 K = GP1*GP2
440 RETURN
450 END
460

```

```

DIMENSION X(25), Y(25), F(25), XO(3), YO(3), FO(3,3), FAC(3,3,4),
1 E(3,3), NIWK(100), WK(120)
DATA NOUT/6/
DATA FAC / 2.5575399 , 1.1562605 , 0.3436082 ,
1.6866026 , 0.7295436 , 0.2116178 ,
0.7374883 , 0.3435128 , 0.0937047 ,
2.6586781 , 1.5227375 , 0.6821580 ,
1.7544727 , 0.8773775 , 0.2846120 ,
0.9337589 , 0.4132084 , 0.1096575 ,
2.5726633 , 1.1580372 , 0.3479553 ,
1.6950045 , 0.7241199 , 0.2139754 ,
0.7366993 , 0.3412316 , 0.0933840 ,
2.3763390 , 1.1071606 , 0.3481252 ,
1.6286966 , 0.7474328 , 0.2177160 ,
0.7119821 , 0.3248314 , 0.0977760 /

```

```

K = 0

```

```

C
DO 100 J=1,5
DO 100 J=1,5
K = K+1

```

```

X(K) = I-1+1./FLOAT(K)
Y(K) = J-1./FLOAT(K)

```

```

100 F(K) = 4.*EXP(-(X(K)**2+Y(K)**2)*.2)

```

```

C
NP = 25

```

```

C
DO 120 I=1,3
XC(I) = I+.5
YO(I) = I-.5

```

```

C
DO 160 MODE=1,3
NIWK = 120
NIWK = 100
CALL LOPE22 (MODE,6,NP,X,Y,F,3,XO,3,YO,IWK,NIWK,WK,NWK,FO,KER)

```

```

C
DC 140 I=1,3
DC 140 J=1,3

```

```

140 E(I,J) = FAC(I,J,MODE)-FU(I,J)

```

```

C
160 WRITE (NOUT,2) MODE,KER,NIWK,NWK
WRITE (NOUT,1) FO,E

```

```

C
IWK(1) = 2
IWK(2) = 3
WK(1) = 2
WK(2) = 1.5
WK(3) = 3.5
WK(4) = 4.+1./21.

```

```

TST00010
TST00020
TST00030
TST00040
TST00050
TST00060
TST00070
TST00080
TST00090
TST00100
TST00110
TST00120
TST00130
TST00140
TST00150
TST00160
TST00170
TST00180
TST00190
TST00200
TST00210
TST00220
TST00230
TST00240
TST00250
TST00260
TST00270
TST00280
TST00290
TST00300
TST00310
TST00320
TST00330
TST00340
TST00350
TST00360
TST00370
TST00380
TST00390
TST00400
TST00410
TST00420
TST00430
TST00440
TST00450
TST00460
TST00470
TST00480

```

Deleted
 (per change)
 attached to
 inside to
 front
 page

THE VALUES OF MODE, KER, NIWK, AND NWK ARE 1 0 79 77

FUNCTION VALUES

2.557540	1.156260	0.343608
1.686603	0.729544	0.211618
0.737488	0.343513	0.093705

DEVIATIONS (THESE VALUES REPRESENT RCUNDOFF ERROR)

-0.000015	0.000042	0.000052
-0.000009	0.000020	0.000050
0.000009	0.000038	0.000009

THE VALUES OF MODE, KER, NIWK, AND NWK ARE 2 0 79 37

FUNCTION VALUES

2.658678	1.522738	0.682158
1.754473	0.877377	0.284612
0.933759	0.413208	0.109657

DEVIATIONS (THESE VALUES REPRESENT RCUNDOFF ERROR)

0.000003	0.000002	-0.000001
0.000001	0.000000	-0.000000
0.000001	-0.000000	0.0

THE VALUES OF MODE, KER, NIWK, AND NWK ARE 3 0 79 104

FUNCTION VALUES

2.572663	1.158037	0.347955
1.695004	0.724120	0.213975
0.736699	0.341232	0.093384

DEVIATIONS (THESE VALUES REPRESENT RCUNDOFF ERROR)

-0.000018	0.000011	0.000047
-0.000015	0.000003	0.000045
-0.000008	0.000040	0.000058

THE VALUES OF MODE, KER, NIWK, AND NWK ARE 3 0 58 76

FUNCTION VALUES

2.376339	1.107161	0.348125
1.626697	0.747433	0.217716
0.711982	0.324831	0.097776

DEVIATIONS (THESE VALUES REPRESENT RCUNDOFF ERROR)

0.000092	-0.000019	-0.000096
0.000033	-0.000010	-0.000066
0.000026	0.000023	0.000017

REFERENCES

1. Hiroshi Akima, "A Method of Bivariate Interpolation and Smooth Surface Fitting for Irregularly Distributed Points", to appear in Transactions on Math Software.
2. R. E. Barnhill, "Representation and Approximation of Surfaces", in Mathematical Software III, pp. 69-120, John R. Rice, ed., Academic Press, 1977.
3. R. E. Barnhill and J. A. Gregory, "Polynomial Interpolation to Boundary Data on Triangles", Math Comp 29 (1975) 726-735.
4. R. Franke, "Locally Determined Smooth Interpolation at Irregularly Spaced Points in Several Variables", JIMA 19 (1977) 471-482.
5. R. Franke, "On the Computation of Optimal Approximations in Sard Corner Spaces $B_{[p,q]}$ ", Technical Report # NPS- 53Fe76121, Naval Postgraduate School, Monterey, CA (to appear in SIAM J. Numer Anal.).
6. R. Franke, "Reproducing Kernel Functions in Sard Corner Spaces", Technical Report # NPR-53-78-001, Naval Postgraduate School, Monterey, CA.
7. C. L. Lawson, "Software for C' Surface Interpolation", in Mathematical Software III, pp. 161-193, John R. Rice, ed., Academic Press, 1977.
8. L. L. Schumaker, "Fitting Surfaces to Scattered Data", in Approximation Theory II, pp. 203-268, edited by G. G. Lorentz, C. K. Chin, and L. L. Schumaker, Academic Press, 1976.

DISTRIBUTION LIST

Defense Documentation Center Cameron Station Alexandria, VA 22314	2
Dudley Knox Library Naval Postgraduate School Monterey, CA 93940	2
Dean of Research Naval Postgraduate School Monterey, CA 93940	2
Department of Mathematics C. O. Wilde, Chairman	1
Professor C. Comstock	1
Professor F. Faulkner	1
Professor A. Schoenstadt	1
Professor R. Franke Naval Postgraduate School Monterey, CA 93940	5
Dr. Richard Lau Office of Naval Research 1030 East Green St. Pasadena, CA 91106	1
Professor R. E. Barnhill Department of Mathematics University of Utah Salt Lake City, UT 84112	1
Professor G. M. Nielson Department of Mathematics Arizona State University Tempe, AZ 85281	1
Chief of Naval Research ATT: Mathematics Program Arlington, VA 22217	2
Rosemary E. Chang Sandia Laboratories Applied Mathematics, Division 2-8235 Livermore, CA 94550	1
Hiroshi Akima Office of Telecommunications Department of Commerce Boulder, CO 80302	1

Professor J. A. Bishop Department of Genetics University of Liverpool P. O. Box 147 Liverpool L693BX, GREAT BRITAIN	1
Dr. Bill Gordon Office of Naval Research/London Box 39 FPO New York, NY 09510	1
W. J. Dejka Naval Ocean Systems Center San Diego, CA 92152	1
Professor R. Wait Department of Computational and Statistical Science University of Liverpool P. O. Box 147 Liverpool, L693BX, GREAT BRITAIN	1
Dr. Herbert Steinberg Mathematical Applications Group, Inc. 3 Westchester Plaza Elmsford, NY 10523	1
Dr. D. Sastry Department of Aeronautical Engineering Indian Institute of Science Bangalore - 560012, INDIA	1
Mark Raterman Silesian University, Institute of Chemistry 9, Szkolna Street, 40-006 Katornice, POLAND	1
Dr. C. L. Lawson Information Systems Division Jet Propulsion Laboratory California Institute of Technology Pasadena, CA 91103	1
Computer Center Professor D. G. Williams	1
Mr. Roger Hilleary Naval Postgraduate School Monterey, CA 93940	1
International Mathematical and Statistical Libraries, Inc. 7500 Bellaire Blvd. Houston, TX 77036	1

U182403

DUDLEY KNOX LIBRARY - RESEARCH REPORTS



5 6853 01060594 2

01060594

MP